

SIEMENS

SINUMERIK

SINUMERIK 840D sl / 828D Расширенное программирование

Руководство по программированию

Действительно для

Система управления
SINUMERIK 840D sl / 840DE sl / 828D

Версия ПО для ЧПУ 4.8 SP3

08/2018
6FC5398-2BP40-6PA2

Предисловие	
Основные указания по безопасности	1
Гибкое программирование ЧПУ	2
Управление файлами и программами	3
Защищенные области	4
Специальные команды перемещения	5
Трансформации координат (фреймы)	6
Трансформации	7
Кинематические цепочки	8
Предотвращение столкновений с кинематическими цепочками	9
Трансформации с кинематическими цепями	10
Коррекции инструмента	11
Параметры траектории	12
Соединения осей	13
Синхронные действия	14
Качание	15
Штамповка и вырубка	16
Шлифование	17
Другие функции	18
Собственные программы обработки резаньем	19
Внешнее программирование циклов	20
Таблицы	21
Приложение	A

Правовая справочная информация

Система предупреждений

Данная инструкция содержит указания, которые Вы должны соблюдать для Вашей личной безопасности и для предотвращения материального ущерба. Указания по Вашей личной безопасности выделены предупреждающим треугольником, общие указания по предотвращению материального ущерба не имеют этого треугольника. В зависимости от степени опасности, предупреждающие указания представляются в убывающей последовательности следующим образом:



ОПАСНО

означает, что непринятие соответствующих мер предосторожности **приводит** к смерти или получению тяжелых телесных повреждений.



ПРЕДУПРЕЖДЕНИЕ

означает, что непринятие соответствующих мер предосторожности **может** привести к смерти или получению тяжелых телесных повреждений.



ОСТОРОЖНО

означает, что непринятие соответствующих мер предосторожности может привести к получению незначительных телесных повреждений.

ВНИМАНИЕ

означает, что непринятие соответствующих мер предосторожности может привести к материальному ущербу.

При возникновении нескольких степеней опасности всегда используется предупреждающее указание, относящееся к наивысшей степени. Если в предупреждении с предупреждающим треугольником речь идет о предупреждении ущерба, причиняемому людям, то в этом же предупреждении дополнительно могут иметься указания о предупреждении материального ущерба.

Квалифицированный персонал

Работать с изделием или системой, описываемой в данной документации, должен только **квалифицированный персонал**, допущенный для выполнения поставленных задач и соблюдающий соответствующие указания документации, в частности, указания и предупреждения по технике безопасности. Квалифицированный персонал в силу своих знаний и опыта в состоянии распознать риски при обращении с данными изделиями или системами и избежать возникающих угроз.

Использование изделий Siemens по назначению

Соблюдайте следующее:



ПРЕДУПРЕЖДЕНИЕ

Изделия Siemens разрешается использовать только для целей, указанных в каталоге и в соответствующей технической документации. Если предполагается использовать изделия и компоненты других производителей, то обязательным является получение рекомендации и/или разрешения на это от фирмы Siemens. Исходными условиями для безупречной и надежной работы изделий являются надлежащая транспортировка, хранение, размещение, монтаж, оснащение, ввод в эксплуатацию, обслуживание и поддержание в исправном состоянии. Необходимо соблюдать допустимые условия окружающей среды. Обязательно учитывайте указания в соответствующей документации.

Товарные знаки

Все наименования, обозначенные символом защищенных авторских прав ©, являются зарегистрированными товарными знаками компании Siemens AG. Другие наименования в данной документации могут быть товарные знаки, использование которых третьими лицами для их целей могут нарушать права владельцев.

Исключение ответственности

Мы проверили содержимое документации на соответствие с описанным аппаратным и программным обеспечением. Тем не менее, отклонения не могут быть исключены, в связи с чем мы не гарантируем полное соответствие. Данные в этой документации регулярно проверяются и соответствующие корректуры вносятся в последующие издания.

Предисловие

Документация по SINUMERIK®

Документация по SINUMERIK подразделяется на следующие категории:

- Общая документация/каталоги
- Документация пользователя
- Документация изготовителя / сервисная документация

Дополнительная информация

По следующему адресу (<https://support.industry.siemens.com/cs/de/en/view/108464614>) находится информация по темам:

- Заказ документации/Обзор документации
- Дополнительные ссылки для загрузки документации
- Использование документации в режиме онлайн (поиск и ознакомление со справочной информацией)

По вопросам технической документации (например, пожелания, исправления) просьба отправить электронное сообщение по следующему адресу (<mailto:docu.motioncontrol@siemens.com>).

MySupport/Документация

По следующему адресу (<https://support.industry.siemens.com/My/ww/en/documentation>) можно найти информацию о самостоятельном составлении документации на основе контента Siemens и ее адаптации к собственной документации по оборудованию.

Обучение

По следующему адресу (<http://www.siemens.com/sitrain>) можно найти информацию по SITRAIN - системе обучения от Siemens по продуктам, системам и решениям по управлению приводом и техники автоматизации.

FAQ

Часто задаваемые вопросы можно найти на страницах Service&Support в поддержке продукта (<https://support.industry.siemens.com/cs/de/en/ps/faq>).

SINUMERIK

Информацию по SINUMERIK можно найти по следующему адресу (<http://www.siemens.com/sinumerik>).

Целевая группа

Настоящая документация предназначена для:

- программистов
- проектировщиков

Преимущества

Руководство по программированию помогает целевой группе в разработке, написании, тестировании и устранении ошибок программ и программных интерфейсов.

Стандартный объем

В настоящем руководстве по программированию описана функциональность стандартного объема. Дополнения и изменения, осуществляемые изготовителем оборудования, документируются изготовителем оборудования.

В СЧПУ могут работать и другие функции, не нашедшие своего отображения в данной документации. Однако претензии по этим функциям не принимаются ни при поставке, ни в случае технического обслуживания.

Кроме этого, данная документация по причине наглядности не содержит всей подробной информации по всем типам продукта и не может предусмотреть каждый мыслимый случай установки, эксплуатации и обслуживания.

Техническая поддержка

Телефоны в конкретных странах для технических консультаций можно найти в Интернете по следующему адресу (<https://support.industry.siemens.com/sc/ww/en/sc/2090>) в разделе «Kontakt».

Информация по структуре и содержанию

Руководство по программированию "Основы" / "Расширенное программирование"

Описания по программированию ЧПУ состоят из двух руководств:

1. Основы

Руководство по программированию "Основы" предназначено для профессиональных операторов станков и предполагает наличие соответствующих знаний в областях сверлильной, фрезерной и токарной обработок. На простых примерах программирования объясняются и известные по DIN 66025 команды и операторы.

2. Расширенное программирование

Руководство по программированию "Расширенное программирование" предназначено для технологов со знанием всех возможностей программирования. СЧПУ SINUMERIK позволяет осуществлять с помощью специального языка программирования программирование комплексной программы обработки детали (к примеру, поверхности свободной формы, координация каналов,...) и облегчает трудоемкое программирование для технологов.

Доступность описанных языковых элементов ЧПУ

Все описанные в настоящем руководстве языковые элементы ЧПУ доступны для SINUMERIK 840D sl. Доступность касательно SINUMERIK 828D см. таблицу "Операторы: Доступность для SINUMERIK 828D (Страница 916)".

Оглавление

	Предисловие.....	3
1	Основные указания по безопасности.....	19
1.1	Общие указания по безопасности.....	19
1.2	Гарантийные обязательства и ответственность за прикладные примеры.....	20
1.3	Промышленная безопасность.....	21
2	Гибкое программирование ЧПУ	23
2.1	Переменные.....	23
2.1.1	Системные данные.....	23
2.1.2	Предопределенные переменные пользователя: R-параметры.....	26
2.1.2.1	Специфические для канала R-параметры (R).....	26
2.1.2.2	Глобальные R-параметры (RG).....	28
2.1.3	Предопределенные переменные пользователя: Link-переменные.....	29
2.1.4	Определение переменных пользователя (DEF).....	31
2.1.5	Переопределение системных данных, пользовательских данных и языковых команд ЧПУ (REDEF).....	38
2.1.6	Атрибут: Инициализирующее значение.....	41
2.1.7	Атрибут: предельные значения (LLI, ULI).....	44
2.1.8	Атрибут: Физическая единица (PHU).....	46
2.1.9	Атрибут: Права доступа (APR, APW, APRP, APWP, APRB, APWB).....	48
2.1.10	Атрибут: Класс данных (DCM, DCI, DCU) - только SINUMERIK 828D.....	54
2.1.11	Обзор определяемых и переопределяемых атрибутов.....	55
2.1.12	Определение и инициализация переменных поля (DEF, SET, REP).....	57
2.1.13	Определение и инициализация переменных поля (DEF, SET, REP): Дополнительная информация.....	61
2.1.14	Типы данных.....	63
2.1.15	Проверка наличия переменной (ISVAR).....	64
2.1.16	Чтение значений атрибута / типа данных (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP).....	65
2.2	Косвенное программирование.....	72
2.2.1	Косвенное программирование адресов.....	72
2.2.2	Косвенное программирование G-команд.....	74
2.2.3	Косвенное программирование атрибутов позиций (GP).....	75
2.2.4	Косвенное программирование строк программы обработки детали (EXECSTRING).....	78
2.3	Функции вычисления.....	80
2.4	Операторы сравнения и логические операторы.....	83
2.5	Коррекция точности при ошибках сравнения (TRUNC).....	85
2.6	Минимум, максимум и диапазон переменных (MINVAL, MAXVAL и BOUND).....	87
2.7	Приоритет операций.....	89
2.8	Возможные преобразования типов.....	90

2.9	Операции со строкой.....	91
2.9.1	Преобразование типов в STRING (AXSTRING)	91
2.9.2	Преобразование типов из STRING (NUMBER, ISNUMBER, AXNAME)	92
2.9.3	Соединение строк (<<)	93
2.9.4	Преобразование в строчные/прописные буквы (TOLOWER, TOUPPER)	94
2.9.5	Определение длины строки (STRLEN)	95
2.9.6	Поиск символа/строки в строке (INDEX, RINDEX, MINDEX, MATCH)	96
2.9.7	Выбор части строки (SUBSTR)	97
2.9.8	Чтение и запись отдельных символов.....	98
2.9.9	Форматирование строки (SPRINT).....	99
2.10	Переходы и ветвления в программе.....	109
2.10.1	Возврат на начало программы (GOTOS).....	109
2.10.2	Переходы на метки перехода (GOTOB, GOTOF, GOTO, GOTOC).....	110
2.10.3	Ветвление программы (CASE ... OF ... DEFAULT ...).....	113
2.11	Повторение блока программы (REPEAT, REPEATB, ENDLABEL, P).....	115
2.12	Управляющие структуры.....	121
2.12.1	Условный оператора и ветвление (IF, ELSE, ENDIF).....	123
2.12.2	Бесконечный программный цикл (LOOP, ENDLOOP).....	124
2.12.3	Счетный цикл (FOR ... TO ..., ENDFOR).....	125
2.12.4	Программный цикл с условием в начале цикла (WHILE, ENDWHILE)	126
2.12.5	Программный цикл с условием на конце цикла (REPEAT, UNTIL)	127
2.12.6	Пример программы со вложенными управляющими структурами.....	127
2.13	Координирующие команды (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM).....	129
2.14	Обработчик прерываний (ASUP).....	134
2.14.1	Функция обработчика прерываний.....	134
2.14.2	Создание обработчика прерываний.....	135
2.14.3	Согласование и запуск обработчика прерываний (SETINT, PRIO, BLSYNC).....	136
2.14.4	Деактивация/повторная активация согласования обработчика прерываний (DISABLE, ENABLE).....	138
2.14.5	Удаление согласования обработчика прерываний (CLRINT).....	138
2.14.6	Быстрый отвод от контура (SETINT LIFTFAST, ALF).....	139
2.14.7	Направление перемещения при быстром отводе от контура	142
2.14.8	Процесс движения для обработчиков прерываний.....	145
2.15	Переход оси, переход шпинделя (RELEASE, GET, GETD).....	146
2.16	Передача оси в другой канал (AXTOCHAN).....	151
2.17	Активация машинных данных (NEWCONF).....	153
2.18	Записать файл (WRITE).....	154
2.19	Удалить файл (DELETE).....	158
2.20	Чтение строк в файле (READ).....	159
2.21	Проверка наличия файла (ISFILE).....	161
2.22	Выгрузка файловой информации (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO).....	162
2.23	Округления с избытком (ROUNDUP).....	164

2.24	Техника подпрограмм.....	165
2.24.1	Общая информация.....	165
2.24.1.1	Подпрограмма.....	165
2.24.1.2	Имена подпрограмм.....	166
2.24.1.3	Вложенность подпрограмм.....	167
2.24.1.4	Маршрут поиска.....	168
2.24.1.5	Формальные и фактические параметры.....	169
2.24.1.6	Передача параметров.....	169
2.24.2	Определение подпрограммы.....	171
2.24.2.1	Подпрограмма без передачи параметров.....	171
2.24.2.2	Подпрограмма с передачей параметров по значению (PROC).....	172
2.24.2.3	Подпрограмма с передачей параметров по ссылке (PROC, VAR).....	174
2.24.2.4	Сохранить модальные функции G (SAVE).....	176
2.24.2.5	Подавление покадровой обработки (SBLOF, SBLON).....	177
2.24.2.6	Подавление актуальной индикации кадра (DISPLOF, DISPLON, ACTBLOCNO).....	183
2.24.2.7	Обозначить подпрограммы с подготовкой (PREPRO).....	186
2.24.2.8	Возврат из подпрограммы M17.....	187
2.24.2.9	Возврат из подпрограммы RET.....	188
2.24.2.10	Параметрируемый возврат из подпрограммы (RET ...).....	189
2.24.2.11	Параметрируемый возврат из подпрограммы (RETB ...).....	195
2.24.3	Вызов подпрограммы.....	199
2.24.3.1	Вызовы подпрограмм без передачи параметров.....	199
2.24.3.2	Вызов подпрограммы с передачей параметров (EXTERN).....	201
2.24.3.3	Число повторений программы (P).....	203
2.24.3.4	Модальный вызов подпрограммы (MCALL).....	205
2.24.3.5	Косвенный вызов подпрограммы (CALL).....	207
2.24.3.6	Косвенный вызов подпрограммы с указанием выполняемого программного блока (CALL BLOCK ... TO ...).....	208
2.24.3.7	Косвенный вызов запрограммированной на языке ISO программы (ISOCALL).....	209
2.24.3.8	Вызов подпрограммы с указанием пути и параметрами (PCALL).....	210
2.24.3.9	Расширить маршрут поиска для вызовов подпрограмм (CALLPATH).....	210
2.24.3.10	Выполнение внешней подпрограммы (840D sl) (EXTCALL).....	212
2.24.3.11	Выполнение внешней подпрограммы (828D) (EXTCALL).....	215
2.25	Техника макросов (DEFINE ... AS).....	220
3	Управление файлами и программами.....	223
3.1	Программная память.....	223
3.1.1	Программная память в ЧПУ.....	223
3.1.2	Внешняя программная память.....	225
3.1.3	Адресация файлов программной памяти.....	227
3.1.4	Маршрут поиска при вызове подпрограммы.....	232
3.1.5	Запрос пути и имени файла.....	233
3.2	Оперативная память (CHANDATA, COMPLETE, INITIAL).....	235
4	Защищенные области.....	239
4.1	Определение защищенных областей (CPROTDEF, NPROTDEF).....	239
4.2	Активация/деактивация защищенных областей (CPROT, NPROT).....	244
4.3	Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI).....	248

5	Специальные команды перемещения.....	259
5.1	Подвод к кодированным позициям (CAC, CIC, CDC, CACP, CACN).....	259
5.2	Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL).....	260
5.3	Соединение сплайнов (SPLINEPATH).....	272
5.4	Включение / выключение сжатия кадра ЧПУ (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF).....	274
5.5	Полиномиальная интерполяция (POLY, POLYPATH, PO, PL).....	275
5.6	Устанавливаемое соотношение траекторий (SPATH, UPATH).....	281
5.7	Измерение с помощью контактного щупа (MEAS, MEAW).....	283
5.8	Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):.....	286
5.9	Специальные функции для OEM-пользователя (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829).....	297
5.10	Уменьшение подачи с замедлением на углах (FENDNORM, G62, G621)	298
5.11	Программируемый критерий окончания движения (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA).....	299
6	Трансформации координат (фреймы).....	303
6.1	Трансформация координат через фрейм-переменные.....	303
6.1.1	Предопределенные фрейм-переменные (\$P_CHBFRAME, \$P_IFRAME, \$P_PFRAME, \$P_ACTFRAME).....	305
6.2	Присвоение значений фреймам.....	309
6.2.1	Присвоение прямых значений (значение оси, угол, масштаб).....	309
6.2.2	Чтение и изменение компонентов фрейма (TR, FI, RT, SC, MI).....	311
6.2.3	Вычисление с фреймами.....	312
6.2.4	Определение фрейм-переменных (DEF FRAME).....	314
6.3	Грубое и точное смещение (CTRANS, CFINE).....	315
6.4	Внешнее смещение нулевой точки (\$AA_ETRANS).....	317
6.5	Установка фактического значения с потерей состояния реферирования (PRESETON).....	319
6.6	Установка фактического значения без потери состояния реферирования (PRESETONS).....	321
6.7	Вычисление фрейма из 3 точек измерения в пространстве (MEAFRAME).....	323
6.8	Глобальные фреймы NCU.....	327
6.8.1	Специфические для канала фреймы (\$P_CHBFR, \$P_UBFR).....	328
6.8.2	Действующие в канале фреймы.....	328
7	Трансформации	335
7.1	Общее программирование типов трансформаций.....	335
7.1.1	Движения ориентации при трансформациях.....	337
7.1.2	Обзор трансформации ориентации TRAORI.....	341
7.2	3-, 4- и 5-осевая трансформация (TRAORI).....	343

7.2.1	Общие связи карданной инструментальной головки.....	343
7.2.2	3-, 4- и 5-осевая трансформация (TRAORI).....	346
7.2.3	Варианты программирования ориентации и первичная установка (ORIRESET).....	347
7.2.4	Программирование ориентации инструмента (A..., B..., C..., LEAD, TILT).....	349
7.2.5	Торцовое фрезерование (A4, B4, C4, A5, B5, C5).....	355
7.2.6	Нулевая точка осей ориентации (ORIWKS, ORIMKS).....	356
7.2.7	Программирование осей ориентации (ORIAxes, ORIVect, ORIEuler, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2).....	358
7.2.8	Программирование ориентации вдоль боковой поверхности конуса (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO).....	360
7.2.9	Задача ориентации двух контактных точек (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=).....	364
7.3	Полиномы ориентации (PO[угол], PO[координата]).....	366
7.4	Вращения ориентации инструмента (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA).....	368
7.5	Ориентации относительно траектории.....	371
7.5.1	Типы ориентаций относительно траектории.....	371
7.5.2	Вращение ориентации инструмента относительно траектории (ORIPATH, ORIPATHS, угол поворота).....	373
7.5.3	Интерполяция вращения инструмента относительно траектории (ORIROTC, THETA).....	374
7.5.4	Сглаживание характеристики ориентации (ORIPATHS A8=, B8=, C8=).....	376
7.6	Сжатие ориентации (COMPON, COMPCURV, COMPCAD, COMPSURF).....	378
7.7	Включение/выключение сглаживания характеристики ориентации (ORISON, ORISOF).....	381
7.8	Кинематическая трансформация.....	383
7.8.1	Включение трансформации с торцевой стороны (TRANSMIT).....	383
7.8.2	Включение трансформации боковой поверхности цилиндра (TRACYL).....	383
7.8.3	Включение косоугольной трансформации с программируемым углом (TRAANG).....	386
7.8.4	Наклонное врезание на шлифовальных станках (G5, G7).....	387
7.9	Включить связанную трансформацию (TRACON).....	389
7.10	Движение "от точки к точке" в декартовой системе координат.....	391
7.10.1	Включение / выключение движения "от точки к точке" в декартовой системе координат (PTP, PTPG0, PTPWOC, CP).....	391
7.10.2	Задание положения шарниров (STAT).....	392
7.10.3	Знак угла между осями (TU).....	396
7.10.4	Пример 1: Движение "от точки к точке" 6-осевого робота с трансформацией ROBX.....	399
7.10.5	Пример 2: Движение "от точки к точке" при базовой 5-осевой трансформации.....	400
7.10.6	Пример 3: PTPG0 и.....	400
7.11	Граничные условия при выборе трансформации.....	402
7.12	Отключение трансформации (TRAFOOF).....	403
8	Кинематические цепочки.....	405
8.1	Удаление компонентов (DELOBJ).....	405
8.2	Определение индекса по имени (NAMETOINT).....	408

9	Предотвращение столкновений с кинематическими цепочками.....	411
9.1	Проверка на соударяемую пару (COLLPAIR).....	412
9.2	Запрос повторного вычисления модели станка для предотвращения столкновений (PROTA).....	413
9.3	Установка состояния защищенной области (PROTS).....	414
9.4	Определение интервала между двумя защищенными областями (PROTD).....	415
10	Трансформации с кинематическими цепями.....	417
10.1	Включение трансформации (TRAFOON).....	417
10.2	Модифицирование трансформации ориентации после замеров станка (CORRTRAFO).....	419
11	Коррекции инструмента.....	427
11.1	Память коррекций.....	427
11.2	Аддитивные коррекции.....	431
11.2.1	Выбор аддитивных коррекций (DL).....	431
11.2.2	Определение значений износа и установочных значений (\$TC_SCPxy[t,d], \$TC_ECPxy[t,d]).....	432
11.2.3	Удаление аддитивных коррекций (DELDL).....	433
11.3	Коррекция инструмента - специальная обработка.....	435
11.3.1	Отражение длин инструмента.....	437
11.3.2	Нормирование знака износа.....	437
11.3.3	Система координат активной обработки (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS).....	438
11.3.4	Длина инструмента и смена плоскостей.....	441
11.4	Коррекция инструмента Online.....	443
11.4.1	Определение функции полинома (FCTDEF).....	443
11.4.2	Онлайн-запись коррекции инструмента, непрерывная (PUTFTOCF).....	444
11.4.3	Дискретная запись онлайн-коррекции инструмента (PUTFTOC).....	445
11.4.4	Включение/выключение онлайн-коррекции инструмента (FTOCON/FTOCOF).....	446
11.5	Коррекция радиуса инструмента 3D.....	447
11.5.1	Коррекция на радиус инструмента 3D для периферийного фрезерования 3D (CUT3DC, CUT3DCD, ISD).....	447
11.5.2	Коррекция на радиус инструмента 3D для торцового фрезерования 3D (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD).....	451
11.5.3	Периферийное фрезерование 3D с учетом ограничивающей поверхности (CUT3DCC, CUT3DCCD).....	457
11.6	Ориентация инструмента (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)....	463
11.7	Присвоение свободных номеров D, номеров резцов.....	469
11.7.1	Присвоение свободных номеров D, номеров резцов (адрес CE).....	469
11.7.2	Присвоение свободных номеров D: Проверка номеров D (CHKDNO).....	469
11.7.3	Присвоение свободных номеров D: Переименование номеров D (GETDNO, SETDNO).....	470
11.7.4	Присвоение свободных номеров D: Определение номера T для заданного номера D (GETACTTD).....	471

11.7.5	Присвоение свободных номеров D: Установка номеров D как недействительных (DZERO).....	471
11.8	Кинематика инструментального суппорта.....	473
11.9	Коррекция длин инструмента для ориентируемого инструментального суппорта (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ).....	479
11.10	Коррекция длин инструмента Online (TOFFON, TOFFOF).....	482
11.11	Изменение данных коррекции для вращающихся инструментов.....	485
11.11.1	Вычисление ориентаций (ORISOLH).....	485
11.11.2	Активация модификации данных коррекции для вращающихся инструментов (CUTMOD, CUTMODK).....	495
11.12	Работа с инструментальным окружением.....	503
11.12.1	Сохранение инструментального окружения (TOOLENV).....	503
11.12.2	Удаление инструментального окружения (DELTOOLENV).....	506
11.12.3	Чтение номеров T, D и DL (GETTENV).....	507
11.12.4	Считывание данных сохраняемого инструментального окружения (\$P_TOOLENVN, \$P_TOOLENV).....	508
11.12.5	Считывание компонентов длины инструмента или компонентов длины (GETTCOR).....	509
11.12.6	Изменение компонентов инструмента (SETTCOR).....	516
11.13	Считывание соотношения длин инструмента L1, L2, L3 с осями координат (LENTOAX).....	529
12	Параметры траектории.....	533
12.1	Тангенциальное управление.....	533
12.1.1	Определение соединения (TANG).....	533
12.1.2	Включение создания промежуточного кадра (TLIFT).....	535
12.1.3	Включение соединения (TANGON).....	536
12.1.4	Выключение соединения (TANGOF).....	537
12.1.5	Удаление соединения (TANGDEL).....	538
12.2	Характеристика подачи (FNORM, FLIN, FCUB, FPO).....	540
12.3	Динамические характеристики.....	545
12.3.1	Режим ускорения (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA).....	545
12.3.2	Управление ускорением для ведомых осей (VELOLIMA, ACCLIMA, JERKLIMA).....	547
12.3.3	Активация спец. для технологии динамических значений (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH).....	549
12.4	Движение с предуплавлением (FFWON, FFWOF).....	551
12.5	Программируемая точность контура (CPRECON, CPRECOF).....	552
12.6	Выполнение программы с памятью предварительной обработки (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)	554
12.7	Определение областей задержки останова (DELAYFSTON, DELAYFSTOF).....	557
12.8	Пропуск места в программе для SERUPRO (IPTRLOCK, IPTRUNLOCK).....	560
12.9	Повторный подвод к контуру (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)	563
12.10	Воздействие на управление движением.....	572
12.10.1	Процентная коррекция рывка (JERKLIM).....	572
12.10.2	Процентная коррекция скорости (VELOLIM).....	573

12.10.3	Пример программы для JERKLIM и VELOLIM.....	575
12.11	Программирование допуска контура/ориентации (CTOL, OTOL, ATOL).....	576
12.12	Параметры смены кадров при активном соединении (CPBC).....	581
13	Соединения осей.....	583
13.1	Буксировка (TRAILON, TRAILOF).....	583
13.2	Таблицы кривых (CTAB).....	588
13.2.1	Определение таблиц кривых (CTABDEF, CATBEND).....	588
13.2.2	Проверка наличия таблицы кривых (CTABEXISTS).....	595
13.2.3	Удаление таблиц кривых (CTABDEL).....	595
13.2.4	Блокировка таблиц кривых от удаления и перезаписи (CTABLOCK, CTABUNLOCK)....	596
13.2.5	Таблицы кривых: Определение свойств таблиц (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD).....	597
13.2.6	Чтение значений таблиц кривых (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX).....	599
13.2.7	Таблицы кривых: Проверка использования ресурсов (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL).....	604
13.3	Осевое соединение по главному значению (LEADON, LEADOF).....	606
13.4	Электронный редуктор (EG).....	612
13.4.1	Определение электронного редуктора (EGDEF).....	612
13.4.2	Включение электронного редуктора (EGON, EGONSYN, EGONSYNE).....	613
13.4.3	Выключение электронного редуктора (EGOFS, EGOFC).....	616
13.4.4	Удаление определения электронного редуктора (EGDEL).....	617
13.4.5	Окружная подача (G95) / электронный редуктор (FPR).....	618
13.5	Синхронный шпиндель.....	619
13.5.1	Синхронный шпиндель: Программирование (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC).....	619
13.6	Базовое соединение осей (CP...).....	630
13.7	Соединение Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS).....	638
14	Синхронные действия.....	641
14.1	Определение синхронного действия.....	641
15	Качание.....	643
15.1	Асинхронное качание (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB).....	643
15.2	Управляемое через синхронные действия качание (OSCILL)	649
16	Штамповка и вырубка.....	657
16.1	Активация/деактивация.....	657
16.1.1	Включение или выключение штамповки и вырубки (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC).....	657
16.2	Автоматическое разделение пути.....	662
16.2.1	Разделение пути для траекторных осей.....	664
16.2.2	Разделение пути для отдельных осей.....	666

17	Шлифование.....	669
17.1	Включение / выключение специфического для шлифования контроля инструмента (TMON, TMOF).....	669
18	Другие функции.....	671
18.1	Осевые функции (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL).....	671
18.2	Переключаемые геометрические оси (GEOAX).....	674
18.3	Осевой контейнер (AXCTSWE, AXCTSWED, AXCTSWEC).....	679
18.4	Ожидание действительной позиции оси (WAITENC).....	681
18.5	Программируемое переключение блоков параметров (SCPARA).....	683
18.6	Проверка имеющейся языковой среды ЧПУ (STRINGIS).....	685
18.7	Интерактивный вызов окон из программы обработки детали (MMC).....	689
18.8	Время выполнения программы / счетчик деталей.....	694
18.8.1	Время выполнения программы.....	694
18.8.2	Счетчики деталей.....	698
18.9	Process DataShare - Вывод на внешнее устройство/файл (EXTOPEN, WRITE, EXTCLOSE).....	700
18.10	Ошибки (SETAL).....	705
18.11	Расширенный останов и отвод (ESR).....	707
18.11.1	Управляемый ЧПУ ESR.....	708
18.11.1.1	Управляемый ЧПУ отвод (POLF, POLFA, POLFMASK, POLFMLIN).....	708
18.11.1.2	Управляемый ЧПУ останов.....	712
18.11.2	Автономный для привода ESR.....	713
18.11.2.1	Конфигурирование автономного останова привода (ESRS).....	713
18.11.2.2	Конфигурирование автономного отвода привода (ESRS).....	713
18.12	Определение заготовки (WORKPIECE).....	715
18.13	Переключение языкового режима (G290, G291).....	720
19	Собственные программы обработки резаньем.....	723
19.1	Функции поддержки для обработки резаньем.....	723
19.2	Создание таблицы контуров (CONTPRON).....	724
19.3	Создание кодированной таблицы контуров (CONTDCON).....	730
19.4	Определить точку пересечения между двумя элементами контура (INTERSEC).....	734
19.5	Покадровая обработка элементов контура таблицы (EXECTAB).....	736
19.6	Вычисление данных окружности (CALCDAT).....	737
19.7	Отключить подготовку контура (EXECUTE).....	739
20	Внешнее программирование циклов.....	741
20.1	Технологические циклы.....	741
20.1.1	Введение.....	741
20.1.2	Обзор по технологиям.....	742
20.1.3	HOLES1 - Ряд отверстий.....	744

20.1.4	HOLES2 - Окружность отверстий.....	745
20.1.5	POCKET3 - Фрезерование прямоугольного кармана.....	747
20.1.6	POCKET4 - Фрезерование кругового кармана.....	749
20.1.7	SLOT1 - Продольный паз.....	752
20.1.8	SLOT2 - кольцевая канавка.....	755
20.1.9	LONGHOLE - Продольный паз.....	757
20.1.10	CYCLE60 - Цикл гравирования.....	759
20.1.11	CYCLE61 - плоское фрезерование.....	762
20.1.12	CYCLE62 - Вызов контура.....	764
20.1.13	CYCLE63 - фрезерование кармана.....	765
20.1.14	CYCLE64 - Предварительное сверление кармана.....	767
20.1.15	CYCLE70 - резьбофрезерование.....	769
20.1.16	CYCLE72 - фрезерование траектории.....	770
20.1.17	CYCLE76 - фрезерование прямоуг. цапфы.....	774
20.1.18	CYCLE77 - фрезерование круговой цапфы.....	776
20.1.19	CYCLE78 - Сверлильное резьбофрезерование.....	778
20.1.20	CYCLE79 - многогранник.....	782
20.1.21	CYCLE81 - сверление, центрование.....	784
20.1.22	CYCLE82 - сверление, цекование.....	785
20.1.23	CYCLE83 - глубокое сверление.....	787
20.1.24	CYCLE84 - нарезание внутренней резьбы без компенсирующего патрона.....	790
20.1.25	CYCLE85 - развертывание.....	793
20.1.26	CYCLE86 - растачивание.....	794
20.1.27	CYCLE92 - отрез.....	796
20.1.28	CYCLE95 - съём припуска вдоль контура.....	797
20.1.29	CYCLE98 - цепочка резьб.....	799
20.1.30	CYCLE99 - нарезание резьбы резцом.....	803
20.1.31	CYCLE435 - установка системы координат правящего инструмента.....	808
20.1.32	CYCLE495 - Профилирование.....	808
20.1.33	CYCLE800 - поворот.....	810
20.1.34	CYCLE801 - Решетка или рамка.....	814
20.1.35	CYCLE802 - любые позиции.....	815
20.1.36	CYCLE830 - Глубокое сверление 2.....	818
20.1.37	CYCLE832 - High Speed Settings.....	825
20.1.38	CYCLE840 - Нарезание внутренней резьбы с компенсирующим патроном.....	829
20.1.39	CYCLE899 - фрезерование открытого паза.....	831
20.1.40	CYCLE930 - выточка.....	834
20.1.41	CYCLE940 - формирование канавки.....	837
20.1.42	CYCLE951 - обработка резанием.....	840
20.1.43	CYCLE952 - прорезание контура.....	843
20.1.44	CYCLE4071 - Продольное шлифование с подачей в точке возврата.....	849
20.1.45	CYCLE4072 - Продольное шлифование с подачей в точке возврата и сигналом отмены.....	850
20.1.46	CYCLE4073 - Продольное шлифование с непрерывной подачей на глубину.....	854
20.1.47	CYCLE4074 - Продольное шлифование с непрерывной подачей на глубину и сигналом отмены.....	855
20.1.48	CYCLE4075 - Плоское шлифование с подачей в точке возврата.....	858
20.1.49	CYCLE4077 - Плоское шлифование с подачей в точке возврата и сигналом отмены.....	861
20.1.50	CYCLE4078 - Плоское шлифование с непрерывной подачей на глубину.....	864
20.1.51	CYCLE4079 - Плоское шлифование с прерывистой подачей на глубину.....	866
20.1.52	GROUP_BEGIN — Начало блока программы.....	868
20.1.53	GROUP_END — Конец блока программы.....	869

20.1.54	GROUP_ADDEND — конец дополнения ввода.....	869
20.1.55	Граничные условия.....	870
20.1.55.1	Технологическое масштабирование внутри масок цикла.....	870
20.2	Измерительные циклы.....	872
21	Таблицы.....	873
21.1	Операторы.....	873
21.2	Операторы: Доступность для SINUMERIK 828D	916
21.2.1	Варианты управления: фрезерование / токарная обработка.....	916
21.2.2	Варианты управления: шлифование.....	944
21.3	Текущий язык в HMI.....	972
A	Приложение.....	973
A.1	Список сокращений.....	973
A.2	Обзор документации.....	982
	Толковый словарь.....	983
	Указатель.....	1007

Основные указания по безопасности

1.1 Общие указания по безопасности

ПРЕДУПРЕЖДЕНИЕ

Опасность для жизни вследствие несоблюдения указаний по безопасности и остаточным рискам

Несоблюдение указаний по безопасности и остаточным рискам, приведенных в соответствующей документации аппаратного обеспечения, может стать причиной тяжелых травм или смерти.

- Строго соблюдайте правила техники безопасности, указанные в документации на аппаратное обеспечение.
- При оценке риска необходимо учитывать остаточные риски.

ПРЕДУПРЕЖДЕНИЕ

Сбои в работе машины вследствие ошибочного или измененного параметрирования

Ошибочное или измененное параметрирование может вызвать нарушения функционирования машин, которые, в свою очередь, могут привести к травмам или к летальному исходу.

- Защищайте параметрирование от некомпетентного вмешательства.
- Устраняйте возможные нарушения функционирования с помощью подходящих мер (например, АВАРИЙНЫЙ ОСТАНОВ или АВАРИЙНОЕ ОТКЛЮЧЕНИЕ).

1.2 Гарантийные обязательства и ответственность за прикладные примеры

Прикладные примеры не носят обязательного характера и не претендуют на полноту в отношении конфигурации и оснащения, а также на универсальность для всех возможных случаев. Прикладные примеры не представляют собой специализированные решения, а лишь предлагают помощь в решении типичных задач.

Вы как пользователь несете ответственность за надлежащую эксплуатацию описанных продуктов. Прикладные примеры не снимают с эксплуатирующего лица ответственности за безопасное обращение при использовании, установке, эксплуатации и техническом обслуживании.

1.3 Промышленная безопасность

Примечание

Промышленная безопасность

Siemens предлагает продукцию и решения с функциями промышленной безопасности, которые обеспечивают безопасную эксплуатацию установок, систем, машин и сетей.

Защита установок, систем, машин и сетей от кибер-угроз предполагает наличие и последовательную поддержку единой концепции промышленной безопасности, соответствующей актуальному техническому уровню. Продукция и решения компании Siemens являются лишь частью такой концепции.

Защита от несанкционированного доступа к установкам, системам, машинам и сетям относится к компетенции заказчика. Подключение систем, машин и компонентов к локальной сети предприятия или Интернету должно осуществляться только при необходимости и с соблюдением соответствующих мер обеспечения безопасности (напр., использование сетевых экранов и сегментация сети).

Дополнительно следует придерживаться рекомендаций Siemens, относящихся к в. у. мерам обеспечения безопасности. Дополнительную информацию о промышленной безопасности можно найти по адресу:

Промышленная безопасность (<http://www.siemens.com/industrialsecurity>)

Безопасность продукции и решений компании Siemens непрерывно совершенствуется. Siemens настоятельно рекомендует устанавливать обновления сразу же после их выхода и всегда использовать только последние версии продуктов. Использование устаревших или более не поддерживаемых версий увеличивает риск кибер-угроз.

Для получения актуальной информации о последних обновлениях можно подписаться на RSS-канал промышленной безопасности Siemens по адресу:

Промышленная безопасность (<http://www.siemens.com/industrialsecurity>)

Дополнительную информацию можно найти в Интернете:

Справочник по проектированию, промышленная безопасность (<https://support.industry.siemens.com/cs/ww/en/view/108862708>)



ПРЕДУПРЕЖДЕНИЕ

Опасные рабочие состояния из-за внесения несанкционированных изменений в программное обеспечение

Внесение несанкционированных изменений в программное обеспечение, например, из-за действия вирусов, троянов, вредоносного ПО или червей, может стать причиной опасных рабочих состояний на установке, и как следствие, привести к смерти, тяжелым травмам и материальному ущербу.

- Постоянно обновляйте ПО.
- Интегрируйте компоненты автоматизации и приводов в единую концепцию промышленной безопасности установки или машины, соответствующую актуальному уровню развития техники.
- В единой концепции промышленной безопасности должны быть учтены все используемые продукты.
- Для защиты файлов на сменных носителях от вредоносного ПО следует использовать соответствующие меры обеспечения безопасности, напр., программы поиска вирусов.
- Необходимо защитить привод от несанкционированных изменений посредством активации функции преобразователя «Защита ноу-хау».

Гибкое программирование ЧПУ

2.1 Переменные

Благодаря использованию переменных, относящихся к системным и пользовательским данным, в первую очередь в комбинации с функциями вычисления и управляющими структурами, можно гибко оформлять программы ЧПУ и циклы.

- **Системные данные**
К системным данным относятся переменные, предопределенные в системе. Эти переменные имеют определенное значение. Они используются системными программами в первую очередь. Пользователь может считывать и записывать эти переменные в программах ЧПУ и циклах. Пример: Характеристики машины, настройки, системные переменные .
Хотя значение системной даты не может редактироваться, пользователь может в известной мере изменить свойства путем переопределения.
См. "Переопределение системных данных, пользовательских данных и языковых команд ЧПУ (REDEF) (Страница 38)"
- **Данные пользователя**
К пользовательским данным относятся определенные пользователем переменные, значение которых задается исключительно пользователем. Система не анализирует эти переменные.
Пользовательские данные делятся на:
 - **Предопределенные переменные пользователя**
Предопределенные переменные пользователя представляют собой уже определенные в системе переменные, количество которых параметрировано в характеристиках машины. Свойства этих переменных могут настраиваться пользователем.
См. "Переопределение системных данных, пользовательских данных и языковых команд ЧПУ (REDEF) (Страница 38)".
 - **Определенных пользователем переменных**
Определенные пользователем переменные представляют собой переменные, определенные пользователем и создаваемые системой для рабочего цикла. Их число, тип данных, видимость и все другие свойства устанавливаются только пользователем.
См. "Определение переменных пользователя (DEF) (Страница 31)"

2.1.1 Системные данные

Системные данные содержат предопределенные в системе переменные, обеспечивающие в программах ЧПУ и циклах доступ к текущему параметрированию СЧПУ, а также состояниям станка, СЧПУ и процессов.

Переменные предварительной обработки

Переменные предварительной обработки представляют собой системные данные, которые считываются и записываются в контексте предварительной обработки, т. е. на момент интерпретации кадра, в котором запрограммирована переменная. Переменная предварительной обработки не создает остановки предварительной обработки.

Переменные главного хода

Переменные главного хода представляют собой системные данные, которые считываются и записываются в контексте главного хода, т. е. на момент исполнения кадра, в котором запрограммирована переменная. Переменными главного хода являются:

- Переменные, которые могут быть запрограммированы в синхронных действиях (чтение/запись)
- Переменные, которые могут быть запрограммированы в программе ЧПУ и вызывают остановку предварительной обработки (чтение/запись)
- Переменные, которые могут быть запрограммированы в программе ЧПУ и получают значение при предварительной обработке, но записываются только на главном ходе (синхронно с главным ходом: только запись)

Систематика префиксов

Для особого обозначения системных данных, перед именем обычно стоит префикс, состоящий из символа \$, одной или двух букв и символа подчеркивания:

\$ + 1. Буква	Объяснение: Тип данных
Данные предварительной обработки (системные данные, считываемые / записываемые при предварительной обработке)	
\$M	Машинные данные ¹⁾
\$S	Установочные данные, защищенные области ¹⁾
\$T	Данные управления инструментом
\$P	Запрограммированные значения
\$C	Переменные оболочек циклов ISO
\$O	Опциональные данные
R	R-параметры (параметры для расчета) ²⁾
Данные главного хода (системные данные, считываемые / записываемые при главном ходе)	
\$\$M	Машинные данные ¹⁾
\$\$S	Установочные данные ¹⁾
\$A	Текущие данные главного хода
\$V	Данные регулятора положения

\$ + 1. Буква	Объяснение: Тип данных
\$R	R-параметры (параметры для расчета) ²⁾
¹⁾ Будут ли машинные и установочные данные обрабатываться как переменные предварительной обработки или переменные главного хода, зависит от того, записываются ли они с одним или двумя символами \$. Форма записи может выбираться свободно в зависимости от приложения. ²⁾ При использовании R-параметра в программе обработки детали / цикле как переменной предварительной обработки, префикс не записывается, например, R10. При использовании в синхронном действии как переменной главного хода в качестве префикса записывается символ \$, например, \$R10.	

2-ая буква	Объяснение: Видимость
N	Глобальная переменная NC (NC)
C	Спец. для канала переменные (Channel)
A	Спец. для оси переменные (Axis)

Граничные условия

Исключение в систематике префиксов

Следующие системные переменные отличаются от в.н. систематики префиксов:

- \$TC_...: 2-ая буква C ссылается не на специфические для канала, а на специфические для зажима инструмента системные переменные (TC = Tool Carrier)
- \$P_ ...: Спец. для канала системные переменные

Использование машинных и установочных данных в синхронных действиях

При использовании машинных и установочных данных в синхронных действиях через префикс можно определить, будут ли машинные или установочные данные считываться/записываться синхронно с предварительной обработкой или главным ходом.

Если данные при обработке не изменяются, то возможно считывание синхронно с предварительной обработкой. Для этого префикс машинных или установочных данных записывается с символом \$:

```
ID=1 WHENEVER $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Если данные при обработке изменяются, то считывание/запись должны выполняться синхронно с главным ходом. Для этого префикс машинных или установочных данных записывается с двумя символами \$:

```
ID=1 WHENEVER $AA_IM[z] < $$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Примечание

Запись машинных и установочных данных

При записи машинных или установочных данных следить за тем, чтобы активный уровень доступа при выполнении программы обработки детали / цикла разрешал бы доступ по записи и чтобы активность данных была бы "IMMEDIATE".

Литература

Полный обзор всех системных переменных можно найти в:
Справочник по параметрированию "Системные переменные"

См. также

Переменные (Страница 23)

2.1.2 Предопределенные переменные пользователя: R-параметры**2.1.2.1 Специфические для канала R-параметры (R)**

Спец. для канала R-параметры это предопределенные переменные пользователя с обозначением R, определенные как поле типа данных REAL. Традиционно, для R-параметров, наряду с написанием с индексом поля, например, R[10], разрешено и написание без индекса поля, например, R10.

При использовании в синхронных действиях должна быть предустановлена буква \$, например, \$R10.

Синтаксис

При использовании в качестве переменной предварительной обработки:

R<n>

R [<выражение>]

При использовании в качестве переменной главного хода:

\$R<n>

\$R [<выражение>]

Объяснение

R:	Идентификатор при использовании в качестве переменной предварительной обработки, например, в программе обработки детали
\$R:	Идентификатор при использовании в качестве переменной главного хода, например, в синхронных действиях

	Тип:	REAL
	Диапазон значений:	При не экспоненциальном представлении: $\pm (0.000\ 0001 \dots 9999\ 9999)$ Примечание: Разрешено макс. 8 десятичных разрядов
		При экспоненциальном представлении: $\pm (1*10^{-300} \dots 1*10^{+300})$ Примечание: • Форма записи: <мантисса>EX<показатель степени>, например 8.2EX-3 • Разрешено макс. 10 символов, включая знак и десятичную запятую.
<n>:	Номер R-параметра	
	Тип:	INT
	Диапазон значений:	0 - MAX_INDEX Указание MAX_INDEX получается из спараметризованного числа R-параметров: MAX_INDEX = (MD28050 \$MN_MM_NUM_R_PARAM) - 1
<выражение>:	Индекс поля В качестве индекса поля может быть выведено любое выражение, пока результат выражения может быть преобразован в тип данных INT (INT, REAL, BOOL, CHAR)	

Пример

Присваивания R-параметрам и использование R-параметров в математических функциях:

Программный код	Комментарий
R0=3.5678	; Присваивание при предварительной обработке
R[1]=-37.3	; Присваивание при предварительной обработке
R3=-7	; Присваивание при предварительной обработке
\$R4=-0.1EX-5	; Присваивание при главном ходе: $R4 = -0.1 * 10^{-5}$
\$R[6]=1.874EX8	; Присваивание при главном ходе: $R6 = 1.874 * 10^8$
R7=SIN(25.3)	; Присваивание при предварительной обработке
R[R2]=R10	; Косвенная адресация через R-параметры
R[(R1+R2)*R3]=5	; Косвенная адресация через матем. выражение
X=(R1+R2)	; Переместить ось X на позицию, получаемую из суммы R1 и R2
Z=SQRT(R1*R1+R2*R2)	; Переместить ось Z на позицию квадратного корня($R1^2 + R2^2$)

См. также

Переменные (Страница 23)

2.1.2.2 Глобальные R-параметры (RG)

Функция

Наряду со специфическими для канала R-параметрами пользователю доступны и глобальные R-параметры. Они существуют внутри СЧПУ однократно и могут считываться/записываться из всех каналов.

Глобальные R-параметры используются, например, для получения информации из одного канала в следующий. Другой пример – глобальные настройки, которые должны обрабатываться для всех каналов, например вылет заготовки из шпинделя.

Считывание и запись глобальных R-параметров осуществляется через пользовательский интерфейс или в программе ЧПУ при предварительной обработке. Использование в синхронных действиях или технологических циклах невозможно.

Примечание

При считывании и записи глобальных R-параметров синхронизация между каналами **не** осуществляется.

Поскольку считывание и запись осуществляются при предварительной обработке, момент времени, в который записанное значение из одного канала активируется в другом канале, не определен.

Пример:

В канале 1 цикл с одним глобальным R-параметром выполняется как счетчик циклов. Канал 2 записывает в этот глобальный R-параметр значение, которое приводит к отмене цикла в канале 1. Однако все циклы, интерпретированные к этому моменту в канале 1 при предварительной обработке, еще выполняются. Число таких циклов не определено и зависит среди прочего от загрузки канала.

Синхронизацию между каналами пользователь должен реализовать самостоятельно на прикладном уровне, например, с помощью меток WAIT!

Синтаксис

Запись в программе ЧПУ

RG [<n>]=<значение>

RG [<выражение>]=<значение>

Чтение в программе ЧПУ

R...=RG [<n>]

R...=RG [<выражение>]

Объяснение

RG:	Стандартное имя адреса ЧПУ для глобальных R-параметров Примечание: Имя адреса ЧПУ можно настраивать через MD15800 \$MN_R_PARAM_NCK_NAME	
<n>:	Номер глобального R-параметра	
	Тип:	INT
	Диапазон значений:	0 ... MAX_INDEX Примечание MAX_INDEX получается из спараметризованного числа глобальных R-параметров: MAX_INDEX = (MD18156 \$MN_MM_NUM_R_PARAM_NCK) - 1
<выражение>:	В качестве индекса поля может быть выведено любое выражение, пока результат выражения может быть преобразован в тип данных INT (INT, REAL, BOOL, CHAR)	
<значение>:	Значение глобального R-параметра	
	Тип:	REAL
	Диапазон значений:	При не экспоненциальном представлении: $\pm (0.000\ 0001 \dots 9999\ 9999)$ Примечание: Разрешено макс. 8 десятичных разрядов При экспоненциальном представлении: $\pm (1 \cdot 10^{-300} \dots 1 \cdot 10^{+300})$ Примечание: - Форма записи: <мантисса>EX<показатель степени>, например 8.2EX-3 - Разрешено макс. 10 символов, включая знак и десятичную запятую.

2.1.3 Предопределенные переменные пользователя: Link-переменные

Через Link-переменные в рамках функции "NCU-Link" возможен обмен циклическими данными между NCU, соединенными друг с другом в одной сети. При этом они обеспечивают спец. для формата данных доступ к памяти Link-переменных. Как размер, так и структура данных памяти Link-переменных, определяется пользователем / изготовителем станка спец. для установки.

Link-переменные это глобальные системные переменные пользователя, которые при сконфигурированной Link-коммуникации могут считываться и записываться всеми NCU структуры связи в программах обработки деталей и циклах. В отличие от глобальных переменных пользователя (GUD), Link-переменные могут использоваться и в синхронных действиях.

На установках без активных NCU-Link, Link-переменные могут использоваться локально для СЧПУ наряду с глобальными переменными пользователя (GUD), как дополнительные глобальные переменные пользователя.

2.1 Переменные

Синтаксис

```

$A_DLB [<индекс>]
$A_DLW [<индекс>]
$A_DLD [<индекс>]
$A_DLR [<индекс>]

```

Значение

\$A_DLB:	Link-переменная для формата данных BYTE (1 байт)	
	Тип данных:	UINT
	Диапазон значений:	0 ... 255
\$A_DLW:	Link-переменная для формата данных WORD (2 байта)	
	Тип данных:	INT
	Диапазон значений:	-32768 ... 32767
\$A_DLD:	Link-переменная для формата данных DWORD (4 байта)	
	Тип данных:	INT
	Диапазон значений:	-2147483648 ... 2147483647
\$A_DLR:	Link-переменная для формата данных REAL (8 байт)	
	Тип данных:	REAL
	Диапазон значений:	$\pm(2,2 \cdot 10^{-308} \dots 1,8 \cdot 10^{+308})$
<индекс>:	Индекс адреса в байтах, рассчитанный от начала памяти Link-переменных	
	Тип данных:	INT
	Диапазон значений:	0 - MAX_INDEX Указание - MAX_INDEX получается из спараметрированного размера памяти Link-переменных: $\text{MAX_INDEX} = (\text{MD18700 } \\$\text{MN_MM_SIZEOF_LINKVAR_DATA}) - 1$ - Могут программироваться только индексы, чтобы адресованные в памяти Link-переменных байты лежали бы в границах одного формата данных $\Rightarrow$ Индекс = $n \cdot \text{байт}$, где $n = 0, 1, 2, \dots$ $\\$A_DLB[i]: i = 0, 1, 2, \dots$ $\\$A_DLW[i]: i = 0, 2, 4, \dots$ $\\$A_DLD[i]: i = 0, 4, 8, \dots$ $\\$A_DLR[i]: i = 0, 8, 16, \dots$

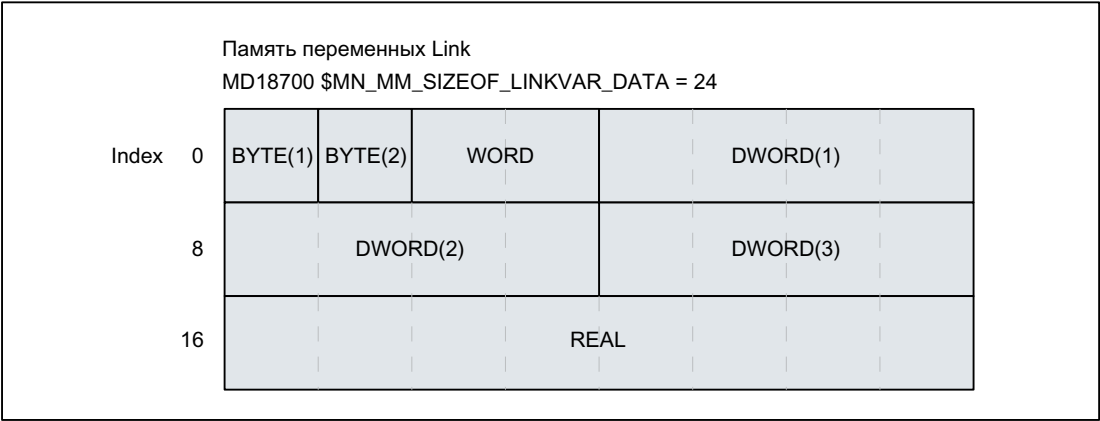
Пример

В системе автоматки имеется 2 NCU (NCU1 и NCU2). К NCU1 подключена ось станка AX2, которая перемещается как Link-ось NCU2.

NCU1 циклически записывает фактическое значение тока (\$VA_CURR) оси AX2 в память Link-переменных. NCU2 циклически считывает переданное через Link-

коммуникацию фактическое значение тока и при превышении предельного значения выводит ошибку 61000.

Структура данных в памяти Link-переменных представлена на следующем рисунке. Фактическое значение тока передается через значение REAL.



УЧПУ1

NCU1 циклически записывает в статическом синхронном действии в такте IPO фактическое значение тока оси AX2 через Link-переменную \$A_DLR[16] в память Link-переменных.

Программный код

N111 IDS=1 WHENEVER TRUE DO \$A_DLR[16]=\$VA_CURR[AX2]

УЧПУ2

NCU2 циклически считывает в статическом синхронном действии в такте IPO фактическое значение тока оси AX2 через Link-переменную \$A_DLR[16] из памяти Link-переменных. Если фактическое значение тока больше 23.0 A, то отображается ошибка 61000.

Программный код

N222 IDS=1 WHEN \$A_DLR[16] > 23.0 DO SETAL(61000)

См. также

Переменные (Страница 23)

2.1.4 Определение переменных пользователя (DEF)

Команда DEF позволяет определять пользовательские переменные (User Data), и присваивать им значения.

Согласно области действия, т. е. области, в которой переменная отображается, существуют следующие категории переменных пользователя:

- **Локальные переменные пользователя (LUD)**
Локальные переменные пользователя (LUD) это переменные, которые определены в программе ЧПУ, которая на момент выполнения не является главной программой. Они создаются при вызове программы ЧПУ и удаляются при сбросе (Reset) конца программы или следующем запуске СЧПУ. Обращение к LUD возможно только внутри программы ЧПУ, в которой они определены.
- **Глобальные программные переменные пользователя (PUD)**
Глобальные программные переменные пользователя (PUD) это переменные пользователя, определенные в используемой в качестве главной программы программе ЧПУ. Они создаются при вызове программы ЧПУ и удаляются при сбросе (Reset) конца программы или следующем запуске СЧПУ. Обращение к PUD возможно в главной программе и во всех подпрограммах.

Примечание

Доступность глобальных программных переменных пользователя (PUD)

Определенные в главной программе глобальные программные переменные пользователя (PUD) доступны в подпрограммах только при следующих установленных машинных данных:

MD11120 \$MN_LUD_EXTENDED_SCOPE = 1

При MD11120 = 0 определенные в главной программе глобальные программные переменные пользователя доступны только в главной программе.

- **Глобальные переменные пользователя (GUD)**
Глобальные переменные пользователя (GUD) – это спец. для ЧПУ или канала глобальные переменные, определенные в блоке данных (SGUD, MGUD, UGUD, GUD4 ... GUD9) и сохраняющиеся после сброса (Reset) конца программы или следующего запуска СЧПУ. Обращение к GUD возможно во всех программах ЧПУ.

Переменные пользователя перед их использованием (чтение / запись) должны быть определены. Соблюдать следующие правила:

- GUD должны быть определены в файле определений, например, _N_DEF_DIR/_N_UGUD_DEF.
- PUD и LUD должны быть определены в разделе описаний программы ЧПУ.
- Описание данных должно выполняться в отдельном кадре.
- Для описания данных может использоваться только один тип данных.
- Для описания данных может быть определено несколько переменных одного типа данных.

Синтаксис

LUD и PUD

```
DEF <тип> <физ_единица> <предельные значения> <имя>[<значение_1>,<значение_2>,<значение_3>]=<иниц_значение>
```


GUD

DEF <область> <остановка предварительной обработки> <права доступа> <класс данных> <тип> <физ_единица> <предельные значения> <имя>[<значение_1>, <значение_2>, <значение_3>]=<иниц_значение>

Объяснение

DEF:	Команда определения переменных пользователя GUD, PUD, LUD	
<область>:	Область действия, релевантно только для GUD:	
	ЧПУ:	Глобальная для ЧПУ переменная пользователя
	CHAN:	Глобальная для канала переменная пользователя
<остановка предварительной обработки>:	Остановка предварительной обработки, релевантно только для GUD (опция)	
	SYNR:	Остановка предварительной обработки при чтении
	SYNW:	Остановка предварительной обработки при записи
	SYNRW:	Остановка предварительной обработки при чтении/записи
<права доступа>:	Степень защиты для чтения / записи GUD через программу ЧПУ или BTSS (опция)	
	APRP <степень защиты>:	Чтение: Программа ЧПУ
	APWP <степень защиты>:	Запись: Программа ЧПУ
	APRB <степень защиты>:	Чтение: OPI
	APWB <степень защиты>:	Запись: OPI
	<степень защиты>:	Диапазон значений: 0 ... 7
	См. "Атрибут: Права доступа (APR, APW, APRP, APWP, APRB, APWB) (Страница 48)"	
<класс данных>:	Назначение класса данных (только SINUMERIK 828D!)	
	DCM:	класс данных M (= Manufacturer)
	DCI:	класс данных I (= Individual)
	DCU:	класс данных U (= User)
	См. "Атрибут: Класс данных (DCM, DCI, DCU) - только SINUMERIK 828D (Страница 54)".	

2.1 Переменные

<тип>:	Тип данных:	
	INT:	Целочисленное значение со знаком
	REAL:	Действительное число (LONG REAL по IEEE)
	BOOL:	Значение истинности TRUE (1) / FALSE (0)
	CHAR:	Символы ASCII
	STRING [<макс.длина>]:	Строка символов определенной длины
	AXIS:	Идентификатор оси/шпинделя
	FRAME:	Геометрические данные для статической трансформации координат
	См. "Типы данных (Страница 63)"	
<физическая единица>:	Физическая единица (опция)	
	PHU <единица>:	физическая единица
	См. "Атрибут: Физическая единица (PHU) (Страница 46)"	
<предельные значения>:	нижнее /верхнее предельное значение (опция)	
	LLI <предельное значение>:	Нижнее предельное значение (lower limit)
	ULI <предельное значение>:	Верхнее предельное значение (upper limit)
	См. "Атрибут: предельные значения (LLI, ULI) (Страница 44)"	
<имя>:	Имя переменной Указание - Макс. 31 символ - Первыми двумя символами должна быть буква и/или символ подчеркивания. - Символ "\$" зарезервирован для системных переменных и не может использоваться.	
[<значение_1>, <значение_2>, <значение_3>]:	Указание размеров поля для 1- до макс. 3-мерных переменных поля (опция) По инициализации переменных поля см. " Определение и инициализация переменных поля (DEF, SET, REP) (Страница 57)"	
<иниц_значение>:	Инициализирующее значение (опция) См. "Атрибут: Инициализирующее значение (Страница 41)" По инициализации переменных поля см. " Определение и инициализация переменных поля (DEF, SET, REP) (Страница 57)"	

Примеры

Пример 1: Определения переменных пользователя в блоке данных для изготовителя станка

Программный код	Комментарий
%_N_MGUD_DEF	; GUD-блок: Изготовитель станка

Программный код	Комментарий
<pre> \$PATH=/_N_DEF_DIR DEF CHAN REAL PHU 24 LLI 0 ULI 10 STROM_1, STROM_2 ; Описание ;Определение двух GUD: STROM_1, STROM_2 ;Область действия: в канале ;Тип данных: REAL ;Остановка предварительной обработки: не запрограммировано => значение по умолчанию = нет остановки предварительной обработки ; Физ. Единица: 24 = [А] ;Предельные значения: Low = 0.0, High = 10.0 ;Права доступа: не запрограммировано => значение по умолчанию = 7 = положение кодового переключателя 0 ;Значение инициализации: не запрограммировано => значение по умолчанию = 0.0 DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 ZEIT_1=12, ZEIT_2=45 ; Описание ;Определение двух GUD: ZEIT_1, ZEIT_2 ;Область действия: По ЧПУ ;Тип данных: REAL ;Остановка предварительной обработки: не запрограммировано => значение по умолчанию = нет остановки предварительной обработки ; Физ. Единица: 13 = [сек] ;Предельные значения: Low = 10.0, High = не запрограммировано => верхняя граница области определения ;Права доступа: ; программа ЧПУ: Запись/чтение = 3 = конечный пользователь ; BTSS: Запись = 0 = Siemens, чтение = 3 = конечный пользователь ;Значение инициализации: ZEIT_1 = 12.0, ZEIT_2 = 45.0 DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER" ; Описание ;Определение одной GUD: GUD5_NAME ;Область действия: По ЧПУ ;Тип данных: STRING, макс. 5 символов ;Остановка предварительной обработки: не запрограммировано => значение по умолчанию = нет остановки предварительной обработки ; Физ. Единица: не запрограммировано => значение по умолчанию = 0 = нет физич. единицы ;Предельные значения: не запрограммировано => границы области определения: Low = 0, High = 255 ;Права доступа: ; программа ЧПУ: Запись/чтение = 3 = конечный пользователь ; BTSS: Запись = 0 = Siemens, чтение = 3 = конечный пользователь ;Значение инициализации: "COUNTER" M30 </pre>	

Пример 2: Глобальные и локальные программные переменные пользователя (PUD / LUD)

Программный код	Комментарий
PROC MAIN	;Главная программа
DEF INT VAR1	; PUD-определение
...	
SUB2	;Вызов подпрограммы
...	
M30	

Программный код	Комментарий
PROC SUB2	;Подпрограмма SUB2
DEF INT VAR2	; LUD-DEFINITION
...	
IF (VAR1==1)	; Чтение PUD
VAR1=VAR1+1	; Чтение и запись PUD
VAR2=1	; Запись LUD
ENDIF	
SUB3	;Вызов подпрограммы
...	
M17	

Программный код	Комментарий
PROC SUB3	;Подпрограмма SUB3
...	
IF (VAR1==1)	; Чтение PUD
VAR1=VAR1+1	; Чтение и запись PUD
VAR2=1	; Ошибка: LUD из SUB2 не известна
ENDIF	
...	
M17	

Пример 3: Определение и использование переменных пользователя типа данных AXIS

Программный код	Комментарий
DEF AXIS ABSZISSE	; 1. Гео-ось
DEF AXIS SPINDLE	; Шпиндель
...	
IF ISAXIS(1)==FALSE GOTOF WEITER	
ABSZISSE = \$P_AXN1	
WEITER:	
...	
SPINDLE=(S1)	; 1. Шпиндель
OVRA[SPINDLE]=80	;Процентровка шпинделя = 80%

Программный код	Комментарий
SPINDLE=(S3)	; 3. Шпиндель

Граничные условия

Глобальные переменные пользователя (GUD)

В рамках определения глобальных переменных пользователя (GUD) должны быть учтены следующие машинные данные:

№	Идентификатор: \$MN_	Объяснение
11140	GUD_AREA_SAVE_TAB	Дополнительная резервная копия для блоков GUD
18118 ¹⁾	MM_NUM_GUD_MODULES	Число файлов GUD в активной файловой системе
18120 ¹⁾	MM_NUM_GUD_NAMES_NCK	Число глобальных имен GUD
18130 ¹⁾	MM_NUM_GUD_NAMES_CHAN	Число спец. для канала имен GUD
18140 ¹⁾	MM_NUM_GUD_NAMES_AXIS	Число спец. для осей имен GUD
18150 ¹⁾	MM_GUD_VALUES_MEM	Место в памяти для глобальных значений GUD
18660 ¹⁾	MM_NUM_SYNACT_GUD_REAL	Число конфигурируемых GUD тип данных REAL
18661 ¹⁾	MM_NUM_SYNACT_GUD_INT	Число конфигурируемых GUD тип данных INT
18662 ¹⁾	MM_NUM_SYNACT_GUD_BOOL	Число конфигурируемых GUD тип данных BOOL
18663 ¹⁾	MM_NUM_SYNACT_GUD_AXIS	Число конфигурируемых GUD тип данных AXIS
18664 ¹⁾	MM_NUM_SYNACT_GUD_CHAR	Число конфигурируемых GUD тип данных CHAR
18665 ¹⁾	MM_NUM_SYNACT_GUD_STRING	Число конфигурируемых GUD тип данных STRING

¹⁾ MD у SINUMERIK 828D только для чтения!

Выходящее за рамки канала использование глобальной переменной пользователя ЧПУ типа данных AXIS

Глобальная переменная пользователя ЧПУ типа данных AXIS, инициализированная при определении в блоке данных с идентификатором оси, может использоваться в различных каналах ЧПУ только в том случае, если ось в этих каналах имеет тот же номер оси канала.

Если это не так, то необходимо загрузить переменную в начале программы ЧПУ или, как в примере ниже, использовать функцию AXNAME(...) (см. "Осевые функции (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Страница 671)").

Программный код	Комментарий
DEF NCK STRING[5] ACHSE="X"	; Определение в блоке данных
...	
N100 AX[AXNAME(ACHSE)]=111 G00	; использование в программе ЧПУ

2.1.5 Переопределение системных данных, пользовательских данных и языковых команд ЧПУ (REDEF)

С помощью команды `REDEF` можно изменять атрибуты системных данных, пользовательских данных и языковых команд ЧПУ. Базовым условием для переопределения является выполнение по времени после соответствующего определения.

Одновременное изменение нескольких атрибутов при переопределении невозможно. Для каждого изменяемого атрибута должен быть запрограммирован отдельный оператор `REDEF`.

Если программируется несколько конкурирующих изменений атрибутов, то всегда активируется последнее изменение.

Сброс величин атрибутов

Измененные с `REDEF` атрибуты для прав доступа и момент инициализации через повторное программирование `REDEF` с последующим именем переменной или языковой команды ЧПУ могут быть сброшены на свое значение по умолчанию:

- Права доступа: Степень защиты 7
- Момент инициализации: нет инициализации или сохранение текущего значения

Переопределяемые атрибуты

См. "Обзор определяемых и переопределяемых атрибутов (Страница 55)".

Локальные переменные пользователя (PUD / LUD)

Переопределения для локальных переменных пользователя (PUD / LUD) невозможны.

Синтаксис

`REDEF <имя> <остановка предварительной обработки>`

`REDEF <имя> <физическая единица>`

`REDEF <имя> <предельные значения>`

`REDEF <имя> <права доступа>`

`REDEF <имя> <момент инициализации>`

`REDEF <имя> <момент инициализации> <иниц_значение>`

`REDEF <имя> <класс данных>`

`REDEF <имя>`

Объяснение

REDEF:	Команда для переопределения определенного атрибута или сброса атрибута "Права доступа" и/или "Момент инициализации" системных переменных, переменных пользователя и языковых команд ЧПУ	
<имя>:	Имя уже определенной переменной или языковой команды ЧПУ	
<VL-Stop>:	Остановка предварительной обработки	
	SYNR:	Остановка предварительной обработки при чтении
	SYNW:	Остановка предварительной обработки при записи
	SYNRW:	Остановка предварительной обработки при чтении/записи
<физическая единица>:	Физическая единица	
	PHU <единица>:	физическая единица
	См. "Атрибут: Физическая единица (PHU) (Страница 46)". Указание Переопределение невозможно для: - Системные переменные - Глобальные данные пользователя (GUD) типов данных: BOOL, AXIS, STRING, FRAME	
<предельные значения>:	нижнее /верхнее предельное значение	
	LLI <предельное значение>:	Нижнее предельное значение (lower limit)
	ULI <предельное значение>:	Верхнее предельное значение (upper limit)
	См. "Атрибут: предельные значения (LLI, ULI) (Страница 44)". Указание Переопределение невозможно для: - Системные переменные - Глобальные данные пользователя (GUD) типов данных: BOOL, AXIS, STRING, FRAME	
<права доступа>:	Права доступа для чтения / записи через программу обработки детали или BTSS	
	APX <степень защиты>:	Выполнение: Языковой элемент ЧПУ
	APRP <степень защиты>:	Чтение: Программа обработки детали
	APWP <степень защиты>:	Запись: Программа обработки детали
	APRB <степень защиты>:	Чтение: OPI
	APWB <степень защиты>:	Запись: OPI
	<степень защиты>:	Диапазон значений: 0 ... 7
	См. "Атрибут: Права доступа (APR, APW, APRP, APWP, APRB, APWB) (Страница 48)".	

<момент инициализации>:	Момент времени повторной инициализации переменной	
	INIPO:	Power On
	INIRE:	Конец главной программы, NC-Reset или Power On
	INICF:	NEWCONF или конец главной программы, сброс ЧПУ или Power On
	PRLOC:	Конец главной программы, NC-Reset после локального изменения или Power On
См. "Атрибут: Инициализирующее значение (Страница 41)".		
<иниц_значение>:	Инициализирующее значение	
	При переопределении инициализирующего значения всегда должен быть указан и момент инициализации (см. <момент инициализации>). См. "Атрибут: Инициализирующее значение (Страница 41)". По инициализации переменных поля см. "Определение и инициализация переменных поля (DEF, SET, REP) (Страница 57)". Указание Переопределение невозможно для системных переменных, кроме установочных данных.	
<класс данных>:	Назначение класса данных (только SINUMERIK 828D!)	
	DCM:	класс данных M (= Manufacturer)
	DCI:	класс данных I (= Individual)
	DCU:	класс данных U (= User)
	См. "Атрибут: Класс данных (DCM, DCI, DCU) - только SINUMERIK 828D (Страница 54)".	

Пример

Переопределение системной переменной \$TC_DPCx в блоке данных для изготовителя станка

Программный код
%_N_MGUD_DEF ; GUD-блок: Изготовитель станка
N100 REDEF \$TC_DPC1 APWB 2 APWP 3
N200 REDEF \$TC_DPC2 PHU 21
N300 REDEF \$TC_DPC3 LLI 0 ULI 200
N400 REDEF \$TC_DPC4 INIPO (100, 101, 102, 103)
N800 REDEF \$TC_DPC1
N900 REDEF \$TC_DPC4
M30

к N100: Право доступа по записи: BTSS = степень защиты 2, программа обработки детали = степень защиты 3

к N200: Физическая единица [%]

к N300: нижнее предельное значение = 0, верхнее предельное значение = 200

к N400: Переменная поля инициализируется при PowerOn с четырьмя значениями.

к N800 / Сброс величин атрибута "Права доступа" и/или "Момент инициализации"
N900

Примечание**Использование файлов ACCESS**

При использовании файлов ACCESS переопределение прав доступа должно быть передано из `_N_MGUD_DEF` в `_N_MACCESS_DEF`.

Граничные условия**Гранулярность**

Переопределение всегда относится к целой, однозначно обозначенной своим именем переменной. Невозможно, например, присваивать отдельным элементам поля переменных поля различные значения атрибутов.

2.1.6 Атрибут: Инициализирующее значение**Определение (DEF) переменных пользователя**

При определении для следующих переменных пользователя может быть задано инициализирующее значение:

- Глобальные переменные пользователя (GUD)
- Глобальные программные переменные пользователя (PUD)
- Локальные переменные пользователя (LUD)

Переопределение (REDEF) системных переменных и переменных пользователя

При переопределении для следующих переменных может быть задано инициализирующее значение:

- Системные данные
 - Установочные данные
- Данные пользователя
 - R-параметры
 - Переменные синхронных действий (\$AC_MARKER, \$AC_PARAM, \$AC_TIMER)
 - GUD синхронных действий (SYG_xy[], где x=R, I, B, A, C, S и y=S, M, U, 4, ..., 9)
 - Параметры EPS
 - Данные инструмента OEM
 - Данные магазина OEM
 - Глобальные переменные пользователя (GUD)

Момент реинициализации

При переопределении можно задать момент времени, когда переменная должна быть реинициализирована, т. е. снова установлена на инициализирующее значение:

- INIPO (Power On)
Реинициализация переменной при PowerOn.
- INIRE (Reset)
Реинициализация переменной при NC-Reset, GPP-Reset, завершении программы обработки детали (M02 / M30) или Power On.
- INICF (NEWCONF)
Реинициализация переменной выполняется при функции "Активация машинных данных" через HMI, команду программы обработки детали NEWCONF или сброс ЧПУ, сброс GPP, завершении программы обработки детали (M02 / M30) или Power On.
- PRLOC (локальное программное изменение)
Реинициализация переменной при NC-Reset, GPP-Reset или завершении программы обработки детали (M02 / M30) только в том случае, если она изменена в рамках текущей программы обработки детали.
Атрибут PRLOC может использоваться только в комбинации с программируемыми установочными данными (см. таблицу ниже).

Таблица 2-1 Программируемые установочные данные

Номер	Идентификатор	G-команда ¹⁾
42000	\$SC_THREAD_START_ANGLE	SF
42010	\$SC_THREAD_RAMP_DISP	DITS / DITE
42400	\$SA_PUNCH_DWELLTIME	PDELAYON
42800	\$SA_SPIND_ASSIGN_TAB	SETMS
43210	\$SA_SPIND_MIN_VELO_G25	G25
43220	\$SA_SPIND_MAX_VELO_G26	G26
43230	\$SA_SPIND_MAX_VELO_LIMS	LIMS

Номер	Идентификатор	G-команда ¹⁾
43300	\$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
43420	\$SA_WORKAREA_LIMIT_PLUS	G26
43430	\$SA_WORKAREA_LIMIT_MINUS	G25
43510	\$SA_FIXED_STOP_TORQUE	FXST
43520	\$SA_FIXED_STOP_WINDOW	FXSW
43700	\$SA_OSCILL_REVERSE_POS1	OSP1
43710	\$SA_OSCILL_REVERSE_POS2	OSP2
43720	\$SA_OSCILL_DWELL_TIME1	OST1
43730	\$SA_OSCILL_DWELL_TIME2	OST2
43740	\$SA_OSCILL_VELO	FA
43750	\$SA_OSCILL_NUM_SPARK_CYCLES	OSNSC
43760	\$SA_OSCILL_END_POS	OSE
43770	\$SA_OSCILL_CTRL_MASK	OSCTRL
43780	\$SA_OSCILL_IS_ACTIVE	OS
43790	\$SA_OSCILL_START_POS	OSB
1) С этой G-командой происходит обращение к установочным данным		

Граничные условия

Инициализирующее значение: Глобальные переменные пользователя (GUD)

- Для глобальных переменных пользователя (GUD) с областью действия ЧПУ в качестве момента инициализации можно задавать только `INIPO` (Power On).
- Для глобальных переменных пользователя (GUD) с областью действия `CHAN` в качестве момента инициализации наряду с `INIPO` (Power On), можно задавать и `INIRE` (Reset) или `INICF` (NEWCONF).
- У глобальных переменных пользователя (GUD) с областью действия `CHAN` и моментом инициализации `INIRE` (Reset) или `INICF` (NEWCONF) при сбросе ЧПУ, сбросе ГПП и "Активация машинных данных" новая инициализация переменных выполняется только в каналах, в которых были запущены названные события.

Инициализирующее значение: Тип данных FRAME

Для переменных типа данных `FRAME` инициализирующее значение не указывается. Переменные типа данных `FRAME` всегда инициализируются не явно с фреймом по умолчанию.

Инициализирующее значение: Тип данных CHAR

Для переменных типа данных `CHAR` вместо кода ASCII (0...255) может быть запрограммирован и соответствующий символ ASCII в кавычках, например, "A"

Инициализирующее значение: Тип данных STRING

У переменных типа данных `STRING` строка символов должна быть заключена в кавычки, например: `...= "MASCHINE_1"`

Инициализирующее значение: Тип данных AXIS

Для переменных типа данных `AXIS` при расширенном написании адреса идентификатор оси должен быть заключен в скобки, например: `...=(X3)`

Инициализирующее значение: Системная переменная

Задача спец. инициализирующих значений пользователя для системных переменных через переопределение невозможна. Инициализирующие значения системных переменных фиксировано заданы системой. Но через переопределение можно изменить момент времени (`INIRE`, `INICF`) реинициализации системной переменной.

Не явное инициализирующее значение: Тип данных AXIS

Для переменных типа данных `AXIS` используется следующее не явное инициализирующее значение:

- Системные данные: "первая гео-ось"
- GUD синхронных действий (обозначение: `SYG_A*`), `PUD`, `LUD`:
идентификатор оси из машинных данных: `MD20082`
`$MC_AXCONF_CHANAX_DEFAULT_NAME`

Не явное инициализирующее значение: Данные инструментов и магазинов

Для данных инструментов и магазинов инициализирующие значения могут быть заданы через следующие машинные данные: `MD17520 $MN_TOOL_DEFAULT_DATA_MASK`

Примечание

Синхронизация

За синхронизацию событий, запускающих реинициализацию глобальной переменной, с чтением этой переменной в другом месте, отвечает исключительно пользователь / изготовитель станка.

См. также

Переменные (Страница 23)

2.1.7 Атрибут: предельные значения (LLI, ULI)

Верхнее и нижнее предельное значение области определения может быть задано только для следующих типов данных:

- `INT`
- `REAL`
- `CHAR`

Определение (DEF) переменных пользователя: Предельные значения и не явные инициализирующие значения

Если при определении переменной пользователя одного из названных выше типов данных явное инициализирующее значение не определяется, то переменная устанавливается на не явное инициализирующее значение типа данных:

- INT: 0
- REAL: 0.0
- CHAR: 0

Если не явное инициализирующее значение выходит за пределы установленной через запрограммированные предельные значения области определения, то переменная инициализируется с предельным значением, наиболее близким к не явному инициализирующему значению:

- Не явное инициализирующее значение < Нижнее предельное значение (LLI) ⇒ Инициализирующее значение = Нижнее предельное значение
- Не явное инициализирующее значение > Верхнее предельное значение (ULI) ⇒ Инициализирующее значение = Верхнее предельное значение

Примеры:

Программный код	Комментарий
DEF REAL GUD1	; Нижнее предельное значение = Граница области определения ; Верхнее предельное значение = Граница области определения ; Инициализирующее значение не запрограммировано ; => Не явное инициализирующее значение = 0.0
DEF REAL LLI 5.0 GUD2	; Нижнее предельное значение = 5.0 ; Верхнее предельное значение = Граница области определения ; => Инициализирующее значение = 5.0
DEF REAL ULI -5 GUD3	; Нижнее предельное значение = Граница области определения ; Верхнее предельное значение = -5.0 ; => Инициализирующее значение = -5.0

Переопределение (REDEF) переменных пользователя: Предельные значения и актуальные фактические значения

Если при переопределении предельных значений переменной пользователя они изменяются таким образом, что актуальное фактическое значение лежит за пределами новой области определения, то выводится ошибка и предельные значения не применяются.

Примечание

Переопределение (REDEF) переменных пользователя

При переопределении предельных значений переменной пользователя следить за непротиворечивым изменением следующих значений:

- Предельные значения
- Фактическое значение
- Инициализирующее значение при переопределении и при автоматической реинициализации из-за INIPO, INIRE или INICF

См. также

Переменные (Страница 23)

2.1.8 Атрибут: Физическая единица (PHU)

Физическая единица может задаваться только для переменных следующих типов данных:

- INT
- REAL

Программируемые физические единицы (PHU)

Физическая единица указывается как число с фиксированной запятой: PHU <единица>

Могут программироваться следующие физические единицы:

<Единица>	Значение	Физическая единица
0	Нет физической единицы	-
1	Линейная или угловая позиция ¹⁾²⁾	[мм], [дюйм], [градус]
2	Линейная позиция ²⁾	[мм], [дюйм]
3	Угловая позиция	[градус]
4	Линейная или угловая скорость ¹⁾²⁾	[мм/мин], [дюймов/мин], [об/мин]
5	Линейная скорость ²⁾	[мм/мин]
6	Угловая скорость	[об/мин]
7	Линейное или угловое ускорение ¹⁾²⁾	[м/сек ²], [дюймов/сек ²], [об/сек ²]
8	Линейное ускорение ²⁾	[м/сек ²], [дюймов/сек ²]

<Единица>	Значение	Физическая единица
9	Угловое ускорение	[об/сек ²]
10	Линейный или угловой рывок ¹⁾²⁾	[м/сек ³], [дюймов/сек ³], [об/сек ³]
11	Линейный рывок ²⁾	[м/сек ³], [дюймов/сек ³]
12	Угловой рывок	[об/сек ³]
13	Время	[сек]
14	Усиление регулятора положения	[16.667/сек]
15	Окружная подача ²⁾	[мм/об], [дюймов/об]
16	Температурная компенсация ¹⁾²⁾	[мм], [дюйм]
18	Сила	[Н]
19	Масса	[кг]
20	Момент инерции ³⁾	[кгм ²]
21	Процент	[%]
22	Частота	[Гц]
23	Напряжение	[В]
24	Ток	[А]
25	Температура	[°С]
26	Угол	[градус]
27	KV	[1000/мин]
28	Линейная или угловая позиция ³⁾	[мм], [дюйм], [градус]
29	Скорость резания ²⁾	[м/мин], [футов/мин]
30	Окружная скорость ²⁾	[м/сек], [футов/сек]
31	Сопротивление	[Ом]
32	Индуктивность	[мГн]
33	Момент вращения ³⁾	[Нм]
34	Постоянная момента вращения ³⁾	[Нм/А]
35	Усиление регулятора тока	[В/А]
36	Усиление регулятора скорости ³⁾	[Нм/(рад*сек)]
37	Скорость	[об/мин]
42	Мощность	[кВт]
43	Ток, низкий	[мкА]
46	Момент вращения, низкий ³⁾	[мкНм]
48	Промилле	-
49	-	[Гц/сек]
65	Расход	[л/мин]
66	Давление	[бар]
67	Объем ³⁾	[см ³]
68	Петлевой коэффициент ³⁾	[мм/(В*мин)]
69	Петлевой коэффициент регулятора усилия	[Н/В]
155	Шаг резьбы ³⁾	[мм/об], [дюймов/об]
156	Изменение шага резьбы ³⁾	[мм/об / об], [дюймов/об / об]
1) Физическая единица зависит от типа оси: линейная или круговая ось		

<Единица>	Значение	Физическая единица
2) Переключение системы единиц G70/G71(дюймовая/метрическая) После переключения основной системы (MD10240 \$MN_SCALING_SYSTEM_IS_METRIC) с G70/ G71 при доступах по записи/чтению к содержащим длины системным переменным и перемен- ным пользователя пересчет значений (фактическое значение, значение по умолчанию и пред- ельные значения не выполняется) G700/G710(дюймовая/метрическая) После переключения основной системы (MD10240 \$MN_SCALING_SYSTEM_IS_METRIC) с G700/ G710 при доступах по записи/чтению к содержащим длины системным переменным и перемен- ным пользователя пересчет значений (фактическое значение, значение по умолчанию и пред- ельные значения выполняется)		
3) Переменная не переводится автоматически в актуальную систему единиц ЧПУ (дюймовая/ метрическая). За перевод отвечает только пользователь / изготовитель станка.		

Примечание**Переполнение уровней из-за преобразования форматов**

Внутренним форматом хранения для переменных пользователя (GUD / PUD / LUD) с содержащими длины физическими единицами является метрический формат. Избыточное использование таких переменных в главном ходе NCK, к примеру, в синхронных действиях, при переключении системы единиц может вызвать перегрузку процессора на уровне интерполятора, ошибка 4240.

Примечание**Совместимость единиц**

При использовании переменных (присваивание, сравнение, вычисление и т. п.), контроль совместимости используемых единиц не осуществляется. За возможно необходимый перевод отвечает только пользователь / изготовитель станка.

См. также

Переменные (Страница 23)

2.1.9 Атрибут: Права доступа (APR, APW, APRP, APWP, APRB, APWB)**Обозначение**

Обозначение атрибутов доступа AP... состоит из:

1. A: **Access** (доступ)
2. P: **Protection** (защита)
3. R / W: **Read / Write** (чтение / запись)
4. P / B: **Program / BTSS** (программа / BTSS)

Права доступа / степени защиты

Правам доступа соответствуют следующие, указываемые при программировании степени защиты:

Право доступа	Степень защиты
Системный пароль	0
Пароль изготовителя станка	1
Сервисный пароль	2
Пароль конечного пользователя	3
Положение кодового переключателя 3	4
Положение кодового переключателя 2	5
Положение кодового переключателя 1	6
Положение кодового переключателя 0	7

Определение (DEF) данных пользователя

Права доступа (APR... / APW...) могут быть определены для следующих данных:

- Глобальные данные пользователя (GUD)

Переопределение (REDEF) системных и пользовательских данных

Права доступа (APR... / APW...) могут быть переопределены для следующих данных:

- Системные данные

- Машинные данные

Примечание

Переопределение прав чтения машинных данных

Степень защиты для чтения машинных данных можно установить только с кодовым словом APR совместно для программы обработки детали BTSS.

При переопределении прав чтения кодовые слова APRP и APRB не поддерживаются и вызывают ошибку 12490 "Право доступа APRP/APRB <уровень защиты> не настроено".

- Установочные данные
- Системная переменная
- Данные процесса
- Данные магазина
- Данные инструмента

- Данные пользователя

- R-параметры
- Переменные синхронных действий (\$AC_MARKER, \$AC_PARAM, \$AC_TIMER)
- GUD синхронных действий (SYG_xy[], где x=R, I, B, A, C, S и y=S, M, U, 4, ..., 9)
- Параметры EPS
- Данные инструмента OEM
- Данные магазина OEM
- Глобальные переменные пользователя (GUD)

Примечание

При переопределении право доступа к переменной между самой низкой степенью защиты 7 и собственной степенью защиты, например, 1 (изготовитель станка) может присваиваться свободно.

Переопределение (REDEF) языковых команд ЧПУ

Право доступа и выполнения (APX) может быть переопределено для следующих языковых команд ЧПУ:

- G-команды / функции перемещения
Литература
Руководство по программированию "Основы"; глава: G-команды / функции перемещения
- Предопределенные функции
Литература
Руководство по программированию "Основы"; глава: Предопределенные функции
- Предопределенные вызовы подпрограмм
Литература
Руководство по программированию "Основы"; глава: Предопределенные вызовы подпрограмм
- Оператор DO для синхронных действий
- Программные идентификаторы циклов
Цикл должен находиться в директории циклов и содержать оператор PROC.

Права доступа касательно программ ЧПУ и циклов (APRP, APWP)

Следствиями различных прав доступа для доступа в программе ЧПУ или цикле являются:

- `APRP 0 / APWP 0`
 - При выполнении программы ЧПУ должен быть установлен системный пароль
 - цикл должен находиться в директории `_N_CST_DIR` (система)
 - для директории `_N_CST_DIR` в MD11160 `$MN_ACCESS_EXEC_CST` право выполнения должно быть установлено на систему
- `APRP 1 / APWP 1` или `APRP 2 / APWP 2`
 - При выполнении программы ЧПУ должен быть установлен пароль изготовителя станка или сервисный пароль
 - цикл должен находиться в директории `_N_CMA_DIR` (изготовитель станка) или `_N_CST_DIR`
 - для директорий `_N_CMA_DIR` или `_N_CST_DIR` в машинных данных MD11161 `$MN_ACCESS_EXEC_CMA` или MD11160 `$MN_ACCESS_EXEC_CST` права выполнения должны быть установлены как минимум на изготовителя станка

2.1 Переменные

- $APRP\ 3 / APWP\ 3$
 - При выполнении программы ЧПУ должен быть установлен пароль конечного пользователя
 - цикл должен находиться в директории $_N_CUS_DIR$ (пользователь) или $_N_CST_DIR$
 - для директорий $_N_CUS_DIR$ или $_N_CMA_DIR$ или $_N_CST_DIR$ в машинных данных MD11162 $\$MN_ACCESS_EXEC_CUS$, MD11161 $\$MN_ACCESS_EXEC_CMA$ или MD11160 $\$MN_ACCESS_EXEC_CST$ права выполнения должны быть установлены как минимум на конечного пользователя
- $APRP\ 4...7 / APWP\ 4...7$
 - При выполнении программы ЧПУ должно быть установлено положение кодового переключателя 3 ... 0
 - цикл должен находиться в директории $_N_CUS_DIR$, $_N_CMA_DIR$ или $_N_CST_DIR$
 - для директорий $_N_CUS_DIR$ или $_N_CMA_DIR$ или $_N_CST_DIR$ в машинных данных MD11162 $\$MN_ACCESS_EXEC_CUS$, MD11161 $\$MN_ACCESS_EXEC_CMA$ или MD11160 $\$MN_ACCESS_EXEC_CST$ права выполнения должны быть установлены как минимум на соответствующее положение кодового переключателя

Права доступа касательно BTSS ($APRB$, $APWB$)

Права доступа ($APRB$, $APWB$) ограничивают в равной мере доступ к системным переменным и переменным пользователя через BTSS для всех системных компонентов (HMI, PLC, внешний процессор, службы EPS и т.д.).

Примечание

Локальные права доступа HMI

При изменениях прав доступа к системным данным необходимо следить за тем, чтобы это не входило в противоречие с установленными через механизмы HMI правами доступа.

Атрибуты доступа APR / APW

По соображениям совместимости, атрибуты APR и APW не явно отображаются на атрибуты $APRP / APRB$ и $APWP / APWB$:

- $APR\ x \Rightarrow APRP\ x\ APRB\ x$
- $APW\ y \Rightarrow APWP\ y\ APWB\ y$

Права доступа через файлы ACCESS

При использовании файлов ACCESS для присваивания прав доступа, переопределения прав доступа для системных данных, данных пользователя и языковых команд ЧПУ могут осуществляться исключительно в файлах ACCESS. Исключением являются глобальные данные пользователя (GUD). Для них переопределение прав доступа должно по-прежнему выполняться в соответствующих файлах определений *_DEF.

Для сквозной защиты доступа необходимо непротиворечивое согласование машинных данных для прав выполнения и защиты доступа соответствующих директорий.

Общий принцип действий:

1. Создание требуемых файлов определений:
 - _N_DEF_DIR/_N_SACCESS_DEF
 - _N_DEF_DIR/_N_MACCESS_DEF
 - _N_DEF_DIR/_N_UACCESS_DEF
2. Параметрирование права записи для файлов определений на требуемое для переопределения значение:
 - MD11170 \$MN_ACCESS_WRITE_SACCESS = <уровень защиты>
 - MD11171 \$MN_ACCESS_WRITE_MACCESS = <уровень защиты>
 - MD11172 \$MN_ACCESS_WRITE_UACCESS = <уровень защиты>
3. Для доступа к защищенным элементам из циклов, необходимо согласовать права выполнения и записи директорий циклов _N_CST_DIR, _N_CMA_DIR и _N_CST_DIR:
Права выполнения
 - MD11160 \$MN_ACCESS_EXEC_CST = <уровень защиты>
 - MD11161 \$MN_ACCESS_EXEC_CMA = <уровень защиты>
 - MD11162 \$MN_ACCESS_EXEC_CUS = <уровень защиты>Права записи
 - MD11165 \$MN_ACCESS_WRITE_CST = <уровень защиты>
 - MD11166 \$MN_ACCESS_WRITE_CMA = <уровень защиты>
 - MD11167 \$MN_ACCESS_WRITE_CUS = <уровень защиты>Право выполнения как минимум должно быть установлено на ту же степень защиты, что и макс. степень защиты используемого элемента.
Право записи как минимум должно быть установлено на ту же степень защиты, что и право выполнения.
4. Права записи локальных директорий циклов HMI должны быть установлены на ту же степень защиты, что и у локальных директорий циклов ЧПУ.
Литература
Руководство по эксплуатации

Вызовы подпрограмм в файлах ACCESS

Для дальнейшего структурирования защиты доступа, в файлах ACCESS возможен вызов и подпрограмм (идентификатор SPF или MPF). При этом подпрограммы получают права выполнения вызывающего файла ACCESS.

Примечание

В файлах ACCESS возможно переопределение только прав доступа. Все другие атрибуты, как и прежде, должны программироваться или переопределяться в соответствующих файлах определений.

См. также

Переменные (Страница 23)

2.1.10 Атрибут: Класс данных (DCM, DCI, DCU) - только SINUMERIK 828D

Для упрощения работы с данными в рамках ввода в эксплуатацию, серийного ввода в эксплуатацию и модернизации станков и серий станков все системные и пользовательские данные ЧПУ разделены на классы данных.

Класс данных	Параметры
S = System	Предоставляемые компанией Siemens системные данные, такие как машинные и установочные данные, стандартные и измерительные циклы, определения (SGUD) и макросы (SMAC) и т.д.
M = Manufacturer (изготовитель станка)	Данные ввода в эксплуатацию для серии станков, такие как циклы изготовителя, определения (MGUD) и макросы (MMAC) и машинные данные, определяющие объем функций станка.
I = Individual (для отдельного станка)	Данные ввода в эксплуатацию для станка, такие как данные компенсации и смещения референтной точки.
U = User (пользователь)	Данные для станка, создаваемые при его эксплуатации, такие как установочные данные, программы обработки детали, циклы пользователя, определения (UGUD) и макросы (UMAC).

Литература:

SINUMERIK 828D Руководство по вводу в эксплуатацию токарных и фрезерных станков; гл.: "Введение и применение классов данных"

Определение (DEF) данных пользователя

Через класс данных файла или директории, в которой определяются данные пользователя, в неявной форме задан класс этих данных. Изменить класс данных невозможно.

Однако при определении (DEF) данных пользователя можно для **значения данных** задать собственный класс, отличающийся от класса этих данных.

Для класса значения данных обязательно следующее:

Приоритет класса значения данных $\leq$ приоритет класса данных

Пример:

Определение глобальных переменных пользователя (GUD), которые описывают измерительный щуп, должно находиться в классе данных M (= Manufacturer), поскольку эти переменные необходимы для выполнения циклов изготовителя. Однако поскольку тип измерительного щупа у разных станков может быть разным, значение данных должно относиться к классу данных I (= Individual).

MGUD.DEF (класс данных M)

...

DEF CHAN DCI INT CALIPER

...

Переопределение (REDEF) системных данных

Класс системных данных можно изменять через переопределение (REDEF).

Переопределение должно осуществляться в файле определений с классом данных S или M.

При использовании ACCESS-файлов переопределения должны осуществляться только внутри ACCESS-файлов.

Соответствующий класс данных машинных, установочных и опциональных данных, а также системных переменных приводится в следующих источниках:

- Справочник по параметрированию "Подробное описание машинных данных", параметр: "Класс"
- Справочник по параметрированию "Системные переменные"

2.1.11 Обзор определяемых и переопределяемых атрибутов

Таблицы ниже показывают, для каких типов данных какие атрибуты могут быть определены (DEF) и/или переопределены (REDEF).

Системные данные

Тип данных	Иниц.значение	Предельные значения	Физ. единица	Права доступа	Класс данных (только 828D)
Машинные данные	---	---	---	REDEF	REDEF
Установочные данные	REDEF	---	---	REDEF	---
FRAME-данные	---	---	---	REDEF	---
Данные процесса	---	---	---	REDEF	---
Комп. погрешности ходового винта (EEC)	---	---	---	REDEF	---
Компенсация провисания (SEC)	---	---	---	REDEF	---

2.1 Переменные

Тип данных	Иниц.значение	Предельные значения	Физ. единица	Права доступа	Класс данных (только 828D)
Компенсация квадрантных ошибок (QEC)	---	---	---	REDEF	---
Данные магазина	---	---	---	REDEF	---
Данные инструмента	---	---	---	REDEF	---
Защищенные области	---	---	---	REDEF	---
Ориентируемые инструментальные суппорта	---	---	---	REDEF	---
Кинематические цепочки	---	---	---	REDEF	---
Защищенные области 3D	---	---	---	REDEF	---
Ограничение рабочего поля	---	---	---	REDEF	---

Данные пользователя

Тип данных	Иниц.значение	Предельные значения	Физ. единица	Права доступа	Класс данных
R-параметры	REDEF	REDEF	REDEF	REDEF	---
Переменные синхронных действий (\$AC_...)	REDEF	REDEF	REDEF	REDEF	---
GUD синхронных действий ((SYG_...))	REDEF	REDEF	REDEF	REDEF	---
Параметры EPS	REDEF	REDEF	REDEF	REDEF	---
Данные инструмента OEM	REDEF	REDEF	REDEF	REDEF	---
Данные магазина OEM	REDEF	REDEF	REDEF	REDEF	---
Глобальные переменные пользователя (GUD)	DEF / REDEF	DEF	DEF	DEF / REDEF	DEF / REDEF
Локальные переменные пользователя (PUD / LUD)	DEF	DEF	DEF	---	---

См. также

Переменные (Страница 23)

2.1.12 Определение и инициализация переменных поля (DEF, SET, REP)

Переменная пользователя может быть определена как 1- до макс. 3-мерное поле (массив):

- 1-мерное: DEF <тип данных> <имя переменной>[<n>]
- 2-мерное: DEF <тип данных> <имя переменной>[<n>, <m>]
- 3-мерное: DEF <тип данных> <имя переменной>[<n>, <m>, <o>]

Примечание

Переменные пользователя типа данных STRING могут быть определены макс. как 2-мерное поле.

Типы данных

Переменные пользователя могут быть определены как поля для следующих типов данных: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

Присваивание значения элементам поля

Значения элементам поля могут присваиваться в следующие моменты времени:

- при определении поля (инициализирующие значения)
- при выполнении программы

При этом значения могут присваиваться через:

- явное указание элемента поля
- явное указание элемента поля как стартового элемента и указание списка значений (SET)
- явное указание элемента поля как стартового элемента и указание значения и частоты его повторения (REP)

Примечание

Переменным пользователя типа данных FRAME не могут быть присвоены инициализирующие значения.

Синтаксис (DEF)

DEF <тип данных> <имя переменной>[<n>, <m>, <o>]

DEF STRING [<длина строки>] <имя переменной>[<n>, <m>]

Синтаксис (DEF...=SET...)

Использование списка значений:

- при определении:
 DEF <тип данных> <имя переменной>[<n>, <m>, <o>] =
 SET (<значение1>, <значение2>, ...)
 равнозначно:
 DEF <тип данных> <имя переменной>[<n>, <m>, <o>] =
 (<значение1>, <значение2>, ...)

Примечание

При инициализации через список значений указание SET является опциональным.

- при присваивании значения:
 <имя переменной>[<n>, <m>, <o>] = SET (<ЗНАЧЕНИЕ1>, <значение2>, ...)

Синтаксис (DEF...=REP...)

Использование значения с повторением

- при определении:
 DEF <тип данных> <имя переменной>[<n>, <m>, <o>] = REP (<значение>)
 DEF <тип данных> <имя переменной>[<n>, <m>, <o>] = REP (<значение> <число
 элементов поля>)
- при присваивании значения:
 <имя переменной>[<n>, <m>, <o>] = REP (<значение>)
 <имя переменной>[<n>, <m>, <o>] = REP (<значение>, <число элементов поля>)

Значение

DEF:	Команда для определения переменных		
<тип данных>:	Тип данных переменных		
	Диапазон значений: • Для системных переменных: BOOL, CHAR, INT, REAL, STRING, AXIS • Для переменных GUD или LUD: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME		
<длина строки>:	Макс. число символов для типа данных STRING		
<имя переменной>:	Имя переменной		
[<n>, <m>, <o>]:	Размеры поля или индексы поля		
<n>:	Размер поля или индекс поля для 1-й размерности		
	Тип:	INT (для системных переменных также AXIS)	
	Диапазон значений:	Макс. размер поля: 65535 Индекс поля: $0 \leq n \leq 65534$	
<m>:	Размер поля или индекс поля для 2-й размерности		
	Тип:	INT (для системных переменных также AXIS)	
	Диапазон значений:	Макс. размер поля: 65535 Индекс поля: $0 \leq m \leq 65534$	

<o>:	Размер поля или индекс поля для 3-й Размерность		
	Тип:	INT (для системных переменных также AXIS)	
	Диапазон значений:	Макс. размер поля: 65535	Индекс поля: $0 \leq o \leq 65534$
SET:	Присваивание значения через указанный список значений		
(<значение1>,<значение2>,...):	Список значений		
REP:	Присваивание значения через указанное <значение>		
<значение>:	Значение, с которым элементы поля должны записываться при инициализации с REP.		
<число элементов поля>:	Число элементов поля, которые должны записываться с указанным <значение>. Для оставшихся элементов поля, в зависимости от момента времени, действует: - Инициализация при определении поля: → Оставшиеся элементы поля записываются с нулем - Присваивание при выполнении программы: → Актуальные значения элементов поля остаются без изменений. Если параметр не запрограммирован, то все элементы поля записываются с <значение>. Если параметр равен нулю, то в зависимости от момента времени действует: - Инициализация при определении поля: → Все элементы предустанавливаются на ноль - Присваивание при выполнении программы: → Актуальные значения элементов поля остаются без изменений.		

Индекс поля

Не явная последовательность элементов поля, к примеру, при присваивании значения через SET или REP, достигается через итерацию индексов поля справа налево.

Пример: Инициализация 3-мерного поля с 24 элементами поля:

```

DEF INT FELD[2,3,4] = REP(1,24)
  FELD[0,0,0] = 1      1. элемент поля
  FELD[0,0,1] = 1      2. элемент поля
  FELD[0,0,2] = 1      3. элемент поля
  FELD[0,0,3] = 1      4. элемент поля
  ...
  FELD[0,1,0] = 1      5. элемент поля
  FELD[0,1,1] = 1      6. элемент поля
  ...
  FELD[0,2,3] = 1      12. элемент поля
  FELD[1,0,0] = 1      13. элемент поля
  FELD[1,0,1] = 1      14. элемент поля

```

```
...
FELD[1,2,3] = 1      24. элемент поля
```

соответственно:

```
FOR n=0 TO 1
  FOR m=0 TO 2
    FOR o=0 TO 3
      FELD[n,m,o] = 1
    ENDFOR
  ENDFOR
ENDFOR
```

Пример: Инициализация целых полей переменных

Актуальные значения см. рисунок.

Программный код

<pre>N10 DEF REAL FELD1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,) N20 FELD1[0,0]=REP(100) N30 FELD1[5,0]=REP(-100) N40 FELD1[0,0]=SET(0,1,2,-10,-11,-12,-20,-20,-20,-30, , , -40,-40,-50,-60,-70) N50 FELD1[8,1]=SET(8.1,8.2,9.0,9.1,9.2)</pre>

Индекс поля

2

1

[1,2]	N10: Инициализация при определении				N20/N30: Инициализация с одинаковым значением				N40/N50: Инициализация с разными значениями		
	0	1	2		0	1	2		0	1	2
0	0	0	0		100	100	100		0	1	2
1	10	11	12		100	100	100		-10	-11	-12
2	20	20	20		100	100	100		-20	-20	-20
3	30	30	30		100	100	100		-30	0	0
4	40	40	40		100	100	100		0	-40	-40
5	0	0	0		-100	-100	-100		-50	-60	-70
6	0	0	0		-100	-100	-100		-100	-100	-100
7	0	0	0		-100	-100	-100		-100	-100	-100
8	0	0	0	-100	-100	-100	-100	8.1	8.2		
9	0	0	0	-100	-100	-100	9.0	9.1	9.2		
		Элементы поля [5,0] до [9,2] были инициализированы со значением по умолчанию (0.0).					Элементы поля [3,1] до [4,0] были инициализированы со значением по умолчанию (0.0). Элементы поля [6,0] до [8,0] остались без изменений.				

Присваивание значения при выполнении программы

При присваивании значения при выполнении программы действуют описанные выше для определения правила. Дополнительно существуют следующие возможности:

- В качестве элементов в списке значений разрешены и выражения.
- Присваивание значения начинается на запрограммированном индексе поля. Благодаря этому возможно и целенаправленное присваивание значений подполям.

Пример:

Программный код	Комментарий
DEF INT FELD[5,5]	; Определение поля
FELD[0,0]=SET(1,2,3,4,5)	; Присваивание значения первым 5 элементам поля [0,0] - [0,4]
FELD[0,0]=SET(1,2, , ,5)	; Присваивание значения с пропуском первым 5 элементам поля [0,0] - [0,4], элементам поля [0,2] и [0,3] = 0
FELD[2,3]=SET(VARIABLE,4*5.6)	; Присваивание значения с переменной и выражением от индекса поля [2,3]: [2,3] = VARIABLE [2,4] = 4 * 5.6 = 22.4

Дополнительная информация (REP)**Инициализация при определении**

- Все или указанное число элементов поля инициализируются с указанным значением (постоянная).
- Переменные типа данных FRAME не могут быть инициализированы.

Пример:

Программный код	Комментарий
DEF REAL varName[10]=REP(3.5,4)	; Определение поля и инициализация элементов поля [0] до [3] со значением 3,5

Присваивание значения при выполнении программы

При присваивании значения при выполнении программы действуют описанные выше для определения правила. Дополнительно существуют следующие возможности:

- В качестве элементов в списке значений разрешены и выражения.
- Присваивание значения начинается на запрограммированном индексе поля. Благодаря этому возможно и целенаправленное присваивание значений подполям.

Примеры:

Программный код	Комментарий
DEF REAL varName[10]	; Определение поля
varName[5]=REP(4.5,3)	; Элементы поля от [5] до [7] = 4,5
R10=REP(2.4,3)	; R-параметры от R10 до R12 = 2,4
DEF FRAME FRM[10]	; Определение поля
FRM[5]=REP(CTrans(X,5))	; Элементы поля от [5] до [9] = CTrans(X,5)

См. также

Определение и инициализация переменных поля (DEF, SET, REP) (Страница 57)

2.1.14 Типы данных

Следующие типы данных имеются в ЧПУ:

Тип данных	Значение	Диапазон значений
INT	Целочисленное значение со знаком	-2147483648 ... +2147483647
REAL	Действительное число (LONG REAL по IEEE)	$\pm(\sim 2,2 \cdot 10^{-308} \dots \sim 1,8 \cdot 10^{+308})$
BOOL	Значение истинности TRUE (1) и FALSE (0)	1, 0
CHAR	Символы ASCII	Код ASCII 0 ... 255
STRING	Строка символов определенной длины	Макс. 200 символов (без специальных символов)
AXIS	Идентификатор оси/шпинделя	Идентификатор оси канала
FRAME	Геометрические данные для статической трансформации координат (смещения, вращения, масштабирования, отражения)	---

Не явные преобразования типов данных

Следующие преобразования типов данных возможны и не явно выполняются при присваиваниях и передачах параметров:

из ↓ / в →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x

x: возможно без ограничений
 O: возможна потеря данных из-за превышения диапазона значений ⇒ ошибка;
 округление: значение после запятой $\geq 0,5 \Rightarrow$ округлять в сторону увеличения, значение после запятой $< 0,5 \Rightarrow$ округлять в сторону уменьшения
 &: значение $\neq 0 \Rightarrow$ TRUE, значение $= 0 \Rightarrow$ FALSE

См. также

Переменные (Страница 23)

2.1.15 Проверка наличия переменной (ISVAR)

С помощью предварительно определенной функции ISVAR можно проверить, известна ли в ЧПУ системная переменная / переменная пользователя (например, машинные данные, установочные данные, системная переменная, общая переменная, например, GUD).

Переменная

Запрашиваемая переменная может иметь следующее строение:

Безразмерная переменная: <переменная>

Одномерная переменная без индекса <переменная>[]
поля:

Одномерная переменная с индексом по- <переменная>[<n>]
ля n:

Двухмерная переменная без индекса по- <переменная>[,]
ля:

Двухмерная переменная с индексами по- <переменная>[<n>,<m>]
ля n и m:

Синтаксис

<результат>=ISVAR (<переменная>[<n>,<m>])

Объяснение

<результат>:	Возвращаемое значение		
	Тип данных:	BOOL	
	Диапазон значений:	1	Имеется переменная
		0	Неизвестная переменная
ISVAR:	Проверяет, известна ли в ЧПУ указанная системная переменная / переменная пользователя.		
<переменная>:	Имя системной переменной / переменной пользователя		
	Тип данных:	STRING	
<n>:	Индекс поля первой размерности (опция)		
	Тип данных:	INT	
<m>:	Индекс поля второй размерности (опция)		
	Тип данных:	INT	

Согласно передаваемому параметру, выполняются следующие проверки:

- Известно ли имя;
- Является ли переменная полем,
- Это одно- или двухмерное поле
- Находится ли соответствующий индекс поля в разрешенном диапазоне.

Если все эти проверки положительные, возвращается TRUE (1).

Если одна из проверок отрицательна, или если обнаруживается синтаксическая ошибка, возвращается FALSE (0).

Примеры

Программный код	Комментарий
DEF INT VAR1 DEF BOOL IS_VAR=FALSE N10 IS_VAR=ISVAR("VAR1")	; IS_VAR в этом случае TRUE.

Программный код	Комментарий
DEF REAL VARARRAY[10,10] DEF BOOL IS_VAR=FALSE N10 IS_VAR=ISVAR("VARARRAY[,]")	; IS_VAR в этом случае TRUE, это двухразмерное поле.
N20 IS_VAR=ISVAR("VARARRAY")	; IS_VAR = TRUE, переменная существует.
N30 IS_VAR=ISVAR("VARARRAY[8,11]")	; IS_VAR = FALSE, индекс поля запрещен.
N40 IS_VAR=ISVAR("VARARRAY[8,8]")	; IS_VAR = FALSE, "]" отсутствует (синтаксическая ошибка).
N50 IS_VAR=ISVAR("VARARRAY[,8]")	; IS_VAR = TRUE, индекс поля разрешен.
N60 IS_VAR=ISVAR("VARARRAY[8,]")	; IS_VAR = TRUE, индекс поля разрешен.

Программный код	Комментарий
DEF BOOL IS_VAR=FALSE N100 IS_VAR=ISVAR("\$MC_GCODE_RESET_VALUES[1]")	; Передаваемый параметр это машинные данные, IS_VAR = TRUE.

Программный код	Комментарий
DEF BOOL IS_VAR=FALSE N10 IS_VAR=ISVAR("\$P_EP")	; IS_VAR в этом случае TRUE.
N20 IS_VAR=ISVAR("\$P_EP[X]")	; IS_VAR в этом случае TRUE.

2.1.16 Чтение значений атрибута / типа данных (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)

С помощью предопределенных функций GETVARPHU, GETVARAP, GETVARLIM и GETVARDFT можно считывать значения атрибутов системных переменных/переменных пользователя, с помощью GETVARTYP тип данных системной переменной/переменной пользователя.

Чтение физической единицы**Синтаксис:**

<результат>=GETVARPHU (<имя>)

Объяснение:

<результат>:	Числовое значение физической единицы		
	Тип данных:	INT	
	Диапазон значений:	см. таблицу в "Атрибут: Физическая единица (PHU) (Страница 46)"	
		В случае ошибки:	
		- 2	Указанное <имя> не присвоено никакому системному параметру и никакой переменной пользователя.
GETVARPHU:	Чтение физической единицы системной переменной/переменной пользователя		
<имя>:	Имя системной переменной/переменной пользователя		
	Тип данных:	STRING	

Пример:

ЧПУ содержит следующую GUD-переменную:

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

Программный код	Комментарий
DEF INT result=0	
result=GETVARPHU("electric"	; Определить физическую единицу GUD-переменной.
)	
IF (result < 0) GOTO error	

Как результат возвращается значение 42. Это соответствует физической единице [кВт].

Примечание

С GETVARPHU можно, например, проверить, имеют ли при присваивании переменной a = b обе переменные ожидаемые физические единицы.

Чтение права доступа**Синтаксис:**

<результат>=GETVARAP (<имя>,<доступ>)

Объяснение:

<результат>:	Степень защиты для указанного <доступа>		
	Тип данных:	INT	
	Диапазон значений:	0 ... 7	См. "Атрибут: Права доступа (APR, APW, APRP, APWP, APRB, APWB) (Страница 48)".
		В случае ошибки:	
		- 1	не поддается записи (имеет значение только при <доступ> "WP" и "WB")
		- 2	Указанное <имя> не присвоено никакому системному параметру и никакой переменной пользователя.
- 3	неправильное значение для <доступа>		
GETVARAP:	Чтение права доступа к системной переменной/переменной пользователя		
<имя>:	Имя системной переменной/переменной пользователя		
	Тип данных:	STRING	
<доступ>:	Тип доступа		
	Тип данных:	STRING	
	Диапазон значений:	"RP"	Чтение через программу обработки детали
		"WP"	Запись через программу обработки детали
		"RB"	Чтение через BTSS
		"WB"	Запись через BTSS

Пример:

Программный код	Комментарий
DEF INT result=0	
result=GETVARAP("\$TC_MAP8", "WB")	; Определить защиту доступа для системного параметра "Позиция магазина" касательно записи через BTSS.
IF (result < 0) GOTO error	

Как результат возвращается значение 7. Это соответствует положению кодового переключателя 0 (= нет защиты доступа).

Примечание

С GETVARAP может быть, например, реализована проверочная программа, проверяющая ожидаемые от приложения права доступа.

Чтение предельных значений**Синтаксис:**

<состояние>=GETVARLIM(<имя>,<предельное значение>,<результат>)

Объяснение:

<состояние>:	Состояние функции		
	Тип данных:	INT	
	Диапазон значений:	1	о. к.
		-1	Предельное значение не определено (для переменных типа AXIS, STRING, FRAME)
		-2	Указанное <имя> не присвоено никаком системному параметру и никакой переменной пользователя.
-3		Неправильное значение для <предельное значение>	
GETVARLIM:	Чтение нижнего/верхнего предельного значения системной переменной/переменной пользователя		
<имя>:	Имя системной переменной/переменной пользователя		
	Тип данных:	STRING	
<предельное значение>:	Указывает, какое предельное значение должно быть выгружено		
	Тип данных:	CHAR	
	Диапазон значений:	"L" :	= нижнее предельное значение
		"U" :	= верхнее предельное значение
<результат>:	Возврат предельного значения		
	Тип данных:	VAR REAL	

Пример:

Программный код	Комментарий
DEF INT state=0	
DEF REAL result=0	
state=GETVARLIM("\$MA_MAX_AX_VELO", "L", r	; Определить нижнее предельное значение для
result)	MD32000 \$MA_MAX_AX_VELO.
IF (result < 0) GOTO error	

Чтение атрибутов / типа данных**Синтаксис:**

<результат>=GETVARDIM(<имя>, индекс)

Объяснение:

<результат>:	Размерность/количество поля <индекс>	
	Тип данных:	INT
GETVARDIM:	Чтение нижнего/верхнего предельного значения системной переменной / переменной пользователя	
<имя>:	Чтение количества элементов поля	
	Тип данных:	STRING
<индекс>:	Номер поля, макс. 3.	
	Тип данных:	INT

Пример:

Программный код	Комментарий
N5 DEF REAL myReal[5,4]	
N10 R1 = GETVATDIM(„myReal“,1)	R1=5
N15 R2 = GETVATDIM(„myReal“,2)	R2=4

Чтение стандартного значения**Синтаксис:**

<состояние>=GETVARDFT (<имя>,<результат>[,<индекс_1>,<индекс_2>,<индекс_3>])

Объяснение:

<состояние>:	Состояние функции		
	Тип данных:	INT	
	Диапазон значений:	1	о. к.
		-1	Стандартное значение отсутствует (например, т.к. тип у <результата> не подходит для <имени>)
		-2	Указанное <имя> не присвоено никаком системному параметру и никакой переменной пользователя.
		-3	Неправильное значение для <индекс_1>, размерность меньше единицы (= не массив = скаляр)
		-4	Неправильное значение для <индекс_2>
		-5	Неправильное значение для <индекс_3>
GETVARDFT:	Чтение стандартного значения системной переменной/переменной пользователя		
<имя>:	Имя системной переменной/переменной пользователя		
	Тип данных:	STRING	
<результат>:	Возврат стандартного значения		
	Тип данных:	VAR REAL (при чтении стандартного значения переменных типов INT, REAL, BOOL, AXIS)	
		VAR STRING (при чтении стандартного значения переменных типов STRING и CHAR)	
		VAR FRAME (при чтении стандартного значения переменных типов FRAME)	
<индекс_1>:	Индекс на первую размерность (как опция)		
	Тип данных:	INT	
	Не запрограммировано означает = 0		
<индекс_2>:	Индекс на вторую размерность (как опция)		
	Тип данных:	INT	
	Не запрограммировано означает = 0		

2.1 Переменные

<индекс_3>:	Индекс на третью размерность (как опция)	
	Тип данных:	INT
	Не запрограммировано означает = 0	

Пример:

Программный код	Комментарий
DEF INT state=0	
DEF REAL resultR=0	; Переменная для записи стандартных значений типов INT, REAL, BOOL, AXIS.
DEF FRAME resultF=0	; Переменная для записи стандартных значений типов FRAME
IF (GETVARTYP("\$MA_MAX_AX_VELO") <> 4) GOTO error	
state=GETVARDEF("\$MA_MAX_AX_VELO", resultR, AXTOINT(X))	; Определить стандартное значение оси "X".
IF (resultR < 0) GOTO error	
IF (GETVARTYP("\$TC_TP8") <> 3) GOTO error	
state=GETVARDEF("\$TC_TP8", resultR)	
IF (GETVARTYP("\$P_UBFR") <> 7) GOTO error	
state=GETVARDEF("\$P_UBFR", resultF)	

Чтение типа данных

Синтаксис:

<результат>=GETVARTYP (<имя>)

Объяснение:

<результат>:	Тип данных указанной системной переменной/переменной пользователя		
	Тип данных:	INT	
	Диапазон значений:	1	= BOOL
		2	= CHAR
		3	= INT
		4	= REAL
		5	= STRING
		6	= AXIS
		7	= FRAME
		В случае ошибки:	
< 0	Указанное <имя> не присвоено никаком системному параметру и никакой переменной пользователя.		
GETVARTYP:	Чтение типа данных системной переменной/переменной пользователя		
<имя>:	Имя системной переменной/переменной пользователя		
	Тип данных:	STRING	

Пример:

Программный код	Комментарий
DEF INT result=0	
DEF STRING name="R"	
result=GETVARTYP(name)	; Определить тип R-параметра.
IF (result < 0) GOTO error	

Как результат возвращается значение 4. Это соответствует типу данных REAL.

2.2 Косвенное программирование

2.2.1 Косвенное программирование адресов

При косвенном программировании адресов расширенный адрес (<индекс>) заменяется на переменную подходящего типа.

Примечание

Косвенное программирование адресов невозможно для:

- N (номер кадра)
- L (подпрограмма)
- Настраиваемых адресов
(к примеру, X[1] вместо X1 запрещено)

Синтаксис

<АДРЕС>[<индекс>]

Значение

<АДРЕС>[...]:	Фиксированный адрес с расширением (индекс)
<индекс>:	Переменная, к примеру, для номера шпинделя, оси, ...

Примеры

Пример 1: Косвенное программирование номера шпинделя

Прямое программирование:

Программный код	Комментарий
S1=300	; Число оборотов 300 об/мин для шпинделя с номером 1.

Косвенное программирование:

Программный код	Комментарий
DEF INT SPINU=1	; Определение переменной типа INT и присваивание значения.
S[SPINU]=300	; Число оборотов 300 об/мин для шпинделя, номер которого сохранен в переменной SPINU (в этом примере это шпиндель с номером 1).

Пример 2: Косвенное программирование оси

Прямое программирование:

Программный код	Комментарий
FA[U]=300	; Подача 300 для оси "U".

Косвенное программирование:

Программный код	Комментарий
DEF AXIS AXVAR2=U	; Определение переменной типа AXIS и присваивание значения.
FA[AXVAR2]=300	; Подача 300 для оси , имя адреса которой сохранено в переменной с именем AXVAR2 .

Пример 3: Косвенное программирование оси

Прямое программирование:

Программный код	Комментарий
\$AA_MM[X]	; Чтение измеренного значения щупа (MCS) оси "X" .

Косвенное программирование:

Программный код	Комментарий
DEF AXIS AXVAR3=X	; Определение переменной типа AXIS и присваивание значения.
\$AA_MM[AXVAR3]	; Чтение измеренного значения щупа (MCS) для оси, имя которой сохранено в переменной AXVAR3 .

Пример 4: Косвенное программирование оси

Прямое программирование:

Программный код
X1=100 X2=200

Косвенное программирование:

Программный код	Комментарий
DEF AXIS AXVAR1 AXVAR2	; Определение двух переменных типа AXIS .
AXVAR1=(X1) AXVAR2=(X2)	; Присваивание имен осей.
AX[AXVAR1]=100 AX[AXVAR2]=200	; Перемещение осей, имена адресов которых сохранены в переменных с именами AXVAR1 и AXVAR2 .

Пример 5: Косвенное программирование оси

Прямое программирование:

Программный код
G2 X100 I20

Косвенное программирование:

Программный код	Комментарий
DEF AXIS AXVAR1=X	; Определение переменной типа AXIS и присваивание значения.
G2 X100 IP[AXVAR1]=20	; Косвенное программирование указания центра для оси , имя адреса которой сохранено в переменной с именем AXVAR1

Пример 6: Косвенное программирование элементов поля

Прямое программирование:

Программный код	Комментарий
DEF INT FELD1[4,5]	; Определение поля 1.

Косвенное программирование:

Программный код	Комментарий
DEFINE DIM1 AS 4	; Для размерностей поля размеры поля должны указываться как фиксированные значения.
DEFINE DIM2 AS 5	
DEF INT FELD[DIM1,DIM2]	
FELD[DIM1-1,DIM2-1]=5	

Пример 7: Косвенный вызов подпрограммы

Программный код	Комментарий
CALL "L" << R10	; Вызов программы, номер которой стоит в R10 (соединение строк).

2.2.2 Косвенное программирование G-команд

Косвенное программирование G-команд обеспечивает эффективное программирование циклов.

Синтаксис

G[<группа>]=<номер>

Объяснение

G[...]:	G-команда с расширением (индекс)	
<группа>:	Параметры индекса: G-группа	
	Тип:	INT
<номер>:	Переменная для номера G-команды	
	Тип:	INT или REAL

Примечание

Как правило, возможно косвенное программирование только не определяющих синтаксис G-команд.

Из определяющих синтаксис G-команд возможна только группа G-команд 1. Определяющие синтаксис G-команды групп G-команд 2, 3 и 4 невозможны.

Примечание

При косвенном программировании G-команды арифметические функции запрещены. Необходимое вычисление номера G-команды должно быть выполнено в отдельной строке программы обработки детали перед косвенным программированием в G-командах.

Примеры

Пример 1: Настраиваемое смещение нулевой точки (группа G-команд 8)

Программный код	Комментарий
N1010 DEF INT INT_VAR	
N1020 INT_VAR=2	
...	
N1090 G[8]=INT_VAR G1 X0 Y0	; G54
N1100 INT_VAR=INT_VAR+1	; Вычисление G-команды
N1110 G[8]=INT_VAR G1 X0 Y0	; G55

Пример 2: Выбор плоскости (группа G-команд 6)

Программный код	Комментарий
N2010 R10=\$P_GG[6]	; Считывание активной G-команды группы G-команд 6
...	
N2090 G[6]=R10	

Литература

Информацию по группам G-команд см.:
Руководство по программированию: основы; глава "Группы G-команд"

2.2.3 Косвенное программирование атрибутов позиций (GP)

Атрибуты позиций, к примеру, инкрементальное или абсолютное программирование позиции оси, в комбинации с кодовым словом GP могут программироваться косвенно как переменные.

Использование

Косвенное программирование атрибутов позиций используется в **циклах замещения**, т.к. здесь существует следующее преимущество по сравнению с программированием атрибутов позиций как кодового слова (к примеру, IC, AC, ...):

Благодаря косвенному программированию как переменной **не нужен** оператор CASE, который выполняет ветвление через все возможные атрибуты позиций.

Синтаксис

<КОМАНДА ПОЗИЦИОНИРОВАНИЯ>[<ось/шпиндель>]=

GP (<позиция>,<атрибут позиции>)

<ось/шпиндель>=GP (<позиция>,<атрибут позиции>)

Значение

<КОМАНДА ПОЗИЦИОНИРОВАНИЯ>[]:	Следующие команды позиционирования могут быть запрограммированы с кодовым словом GP: POS, POSA, SPOS, SPOSA Кроме этого возможны: - Все имеющиеся в канале идентификаторы осей/шпинделей: <ось/шпиндель> - Переменный идентификатор оси/шпинделя AX
<ось/шпиндель>:	Ось/шпиндель, которая должна быть позиционирована
GP () :	Кодовое слово для позиционирования
<позиция>:	Параметр 1 Позиция оси/шпинделя как постоянная или переменная
<атрибут позиции>:	Параметр 2 Атрибут позиции (к примеру, режим подвода к позиции) как переменная (к примеру, \$P_SUB_SPOSMODE) или как кодовое слово (IC, AC, ...)

Выводимые переменными величины имеют следующее значение:

Величина	Значение	Допускается для:
0	Нет изменений атрибута позиции	
1	AC	POS, POSA, SPOS, SPOSA, AX, адрес оси
2	IC	POS, POSA, SPOS, SPOSA, AX, адрес оси
3	DC	POS, POSA, SPOS, SPOSA, AX, адрес оси
4	ACP	POS, POSA, SPOS, SPOSA, AX, адрес оси
5	ACN	POS, POSA, SPOS, SPOSA, AX, адрес оси
6	OC	-
7	PC	-
8	DAC	POS, POSA, AX, адрес оси
9	DIC	POS, POSA, AX, адрес оси
10	RAC	POS, POSA, AX, адрес оси
11	RIC	POS, POSA, AX, адрес оси
12	CAC	POS, POSA
13	CIC	POS, POSA
14	CDC	POS, POSA
15	CACP	POS, POSA
16	CACN	POS, POSA

Пример

При активном соединении синхронных шпинделей между ходовым винтом S1 и ведомым шпинделем S2 через команду SPOS в главной программе вызывается следующий цикл замещения для позиционирования шпинделей.

Позиционирование осуществляется через оператора в N2230:

```
SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE) SPOS[2]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
```

Позиция подвода считывается из системной переменной \$P_SUB_SPOSIT, режим подвода к позиции - из системной переменной \$P_SUB_SPOSMODE.

Программный код	Комментарий
N1000 PROC LANG_SUB DISPL0F SBLOF	
...	
N2100 IF(\$P_SUB_AXFCT==2)	
N2110	; Замещение команды SPOS / SPOSA / M19 при активном соединении синхронных шпинделей
N2185 DELAYFSTON	; Начало области задержки останова
N2190 COUPOF(S2,S1)	; Деактивировать соединение синхронных шпинделей
N2200	; Позиционирование ведущего и ведомого шпинделя
N2210 IF(\$P_SUB_SPOS==TRUE) OR (\$P_SUB_SPOSA==TRUE)	
N2220	; Позиционировать шпиндель со SPOS:
N2230 SPOS[1]=GP(\$P_SUB_SPOSIT, \$P_SUB_SPOSMODE)	
SPOS[2]=GP(\$P_SUB_SPOSIT, \$P_SUB_SPOSMODE)	
N2250 ELSE	
N2260	; Позиционировать шпиндель с M19:
N2270 M1=19 M2=19	; Позиционирование ведущего и ведомого шпинделя
N2280 ENDIF	
N2285 DELAYFSTOF	; Конец области задержки останова
N2290 COUPON(S2,S1)	; Активировать соединение синхронных шпинделей
N2410 ELSE	
N2420	; Опрос на предмет других замещений
...	
N3300 ENDIF	
...	
N9999 RET	

Граничные условия

- В синхронных действиях косвенное программирование атрибутов позиций невозможно.

Литература

Описание функций "Основные функции"; ГПП, канал, программный режим, параметры Reset (K1),
глава: Замещение функций ЧПУ через подпрограммы

2.2.4 Косвенное программирование строк программы обработки детали (EXECSTRING)

С помощью команды программы обработки детали EXECSTRING можно выполнить созданную прежде переменную String как строку программы обработки детали.

Синтаксис

EXECSTRING программируется в отдельной строке программы обработки детали:
EXECSTRING (<переменная String>)

Значение

EXECSTRING:	Команда для выполнения переменной String как строки программы обработки детали
<переменная String>:	Переменная типа STRING, содержащая саму исполняемую строку программы обработки детали

Примечание

С помощью EXECSTRING могут устанавливаться, за исключением управляющих структур (Страница 121), все конструкции программы обработки детали, которые могут быть запрограммированы **в программном блоке программы обработки детали**.
Исключением являются операторы PROC и DEF, а также использование в файлах INI и DEF.

Пример

Программный код	Комментарий
N100 DEF STRING[100] MY_BLOCK	; Определение переменной String для записи исполняемой строки программы обработки детали.
N110 DEF STRING[10] MFCT1="M7"	
...	
N200 EXECSTRING(MFCT1 << "M4711")	; Выполнить строку программы обработки детали "M7 M4711".
...	
N300 R10=1	
N310 MY_BLOCK="M3"	
N320 IF(R10)	

Программный код	Комментарий
N330 MY_BLOCK = MY_BLOCK << MFCT1 N340 ENDIF N350 EXECSTRING(MY_BLOCK)	; Выполнить строку программы обработки детали "M3 M7".

2.3 Функции вычисления

Оператор / функция вычисления	Значение
+	Сложение
-	Вычитание
*	Умножение
/ ¹⁾	Деление ¹⁾
DIV ¹⁾	Целочисленное деление ¹⁾
MOD ¹⁾	Деление по модулю (выдает остаток целочисленного деления) ¹⁾
:	Связывающий оператор для переменных FRAME
SIN ()	Синус
COS ()	Косинус
TAN ()	Тангенс
ASIN ()	Арксинус
ACOS ()	Арккосинус
ATAN2 (,) ¹⁾	Арктангенс2 ¹⁾
SQRT ()	Квадратный корень
ABS ()	Значение
POT ()	2-я степень (квадрат)
TRUNC ()	Целая часть Настраиваемая точность для команд сравнения с TRUNC (см. "Коррекция точности при ошибках сравнения (TRUNC) (Страница 85)")
ROUND ()	Округление до целого
LN ()	Натуральный логарифм
EXP ()	Показательная функция
MINVAL ()	Меньшее значение двух переменных (см. "Минимум, максимум и диапазон переменных (MINVAL, MAXVAL и BOUND) (Страница 87)")
MAXVAL ()	Большее значение двух переменных (см. "Минимум, максимум и диапазон переменных (MINVAL, MAXVAL и BOUND) (Страница 87)")
BOUND ()	Значение переменной, лежащее в определенном диапазоне значений (см. "Минимум, максимум и диапазон переменных (MINVAL, MAXVAL и BOUND) (Страница 87)")
CTrans ()	Смещение
CROT ()	Вращение
CSCALE ()	Изменение масштаба
CMIRROR ()	Отражение
1) См. абзац "Примеры"	

Программирование

Для функций вычисления действует обычное математическое представление. Приоритеты в выполнении устанавливаются посредством круглых скобок. Для тригонометрических и их обратных функций действует указание градуса (прямой угол = 90°).

Примеры

Деление: /

(тип REAL) = (тип INT или тип REAL) / (тип INT или тип REAL);

Пример: $3 / 4 = 0.75$

Целочисленное деление: DIV

(тип INT) = (тип INT или REAL) / (тип INT или REAL);

Пример: $7 \text{ DIV } 4.1 = 1$

Деление по модулю (выдает остаток целочисленного деления): MOD

(тип REAL) = (тип INT или REAL) MOD (тип INT или REAL);

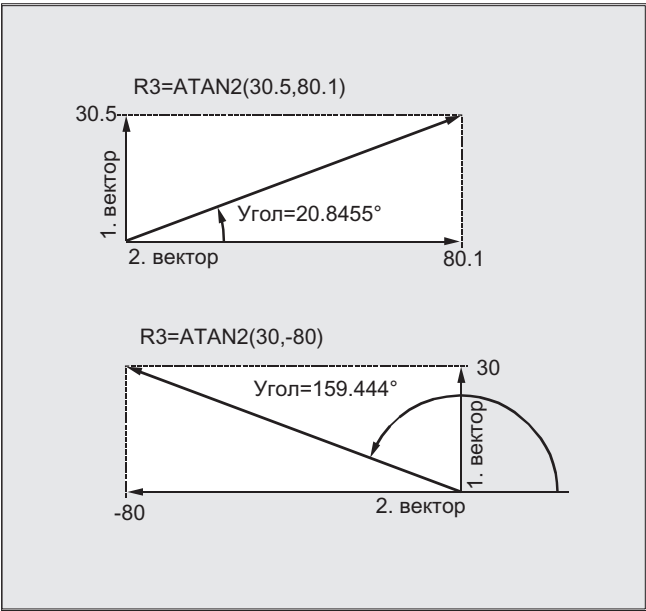
Пример: $7 \text{ MOD } 4.1 = 2.9$

Арктангенс2: ATAN2

Функция вычисляет ATAN2 из двух расположенных вертикально друг на друге векторов угол суммарного вектора.

Результат находится в диапазоне четырех квадрантов ($-180^\circ < 0 < +180^\circ$).

Основой для угловой корреляции всегда является 2-е значение в положительном направлении.



Примеры программирования

Программный код	Комментарий
R1=R1+1	; Новое R1 = старое R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; Вычисление по правилам арифметики.
R14=(R1+R2)*R3	; Сначала вычисляются выражения в скобках.
R15=SQRT(POT(R1)+POT(R2))	; Сначала раскрываются внутренние скобки: R15 = квадратный корень((R1^2 + R2^2))
RESFRAME=FRAME1:FRAME2	; FRAME-соединение со связывающим оператором
FRAME3=CTTRANS(...):CROT(...)	Присваивание значения FRAME-компоненту

2.4 Операторы сравнения и логические операторы

Операции сравнения могут использоваться, к примеру, для формулировки условия перехода. При этом возможно сравнение и сложных выражений.

Операции сравнения могут использоваться для переменных типа `CHAR`, `INT`, `REAL` и `BOOL`. Для типа `CHAR` сравнивается кодовое значение.

Для типов `STRING`, `AXIS` и `FRAME` возможны: `==` и `<>`, которые для операций типа `STRING` могут применяться и в синхронных действиях.

Результат сравнительных операций всегда имеет тип `BOOL`.

Логические операторы служат для связи значений истинности.

Логические операции могут применяться только для переменных типа `BOOL`. Через внутреннее преобразование типов они могут использоваться и для типов данных `CHAR`, `INT`, и `REAL`.

Для логических (булевых) операций для типов данных `BOOL`, `CHAR`, `INT` и `REAL` действует:

- 0 соответствует: `false`
- Отлично от 0 соответствует: `TRUE`

Побитовые логические операторы

С помощью переменных типа `CHAR` и `INT` могут выполняться и побитовые логические операции. При необходимости преобразование типов выполняется автоматически.

Программирование

Оператор сравнения	Значение
<code>==</code>	равно
<code><></code>	не равно
<code>></code>	больше
<code><</code>	меньше
<code>>=</code>	больше или равно
<code><=</code>	меньше или равно

Логический оператор	Значение
<code>AND</code>	И
<code>OR</code>	ИЛИ
<code>NOT</code>	Отрицание
<code>XOR</code>	исключающее ИЛИ

Побитовый логический оператор	Значение
<code>B_AND</code>	побитовый И
<code>B_OR</code>	побитовый ИЛИ
<code>B_NOT</code>	побитовое отрицание
<code>B_XOR</code>	побитовое исключяющее ИЛИ

Примечание

В арифметических выражениях посредством круглых скобок можно устанавливать последовательность выполнения всех операторов, отличную от обычных правил приоритета.

Примечание

Между ЛОГИЧЕСКИМИ операндами и операторами необходимо вставлять пробелы.

Примечание

Оператор `B_NOT` относится только к одному операнду. Он стоит после оператора.

Примеры

Пример 1: Операторы сравнения

```
IF R10>=100 GOTO ZIEL
```

или

```
R11=R10>=100
```

```
IF R11 GOTO ZIEL
```

Результат сравнения `R10>=100` сначала буферизируется в `R11`.

Пример 2: Логические операторы

```
IF (R10<50) AND ($AA_IM[X]>=17.5) GOTO ZIEL
```

или

```
IF NOT R10 GOTOB START
```

`NOT` относится только к одному операнду.

Пример 3: Побитовые логические операторы

```
IF $MC_RESET_MODE_MASK B_AND 'B10000' GOTO ACT_PLANE
```

2.5 Коррекция точности при ошибках сравнения (TRUNC)

Команда TRUNC отрезает умноженный на коэффициент точности операнд.

Настраиваемая точность для команд сравнения

Внутреннее отображение данных программы обработки детали типа REAL осуществляется в формате IEEE с 64 битами. Из-за такого формата десятичные числа могут отображаться неточно, что при сравнении с идеально вычисленными значениями может привести к нежелательным результатам.

Относительное равенство

Для того, чтобы обусловленные форматом отображения неточности, не сказывались отрицательно на ходе программы, команды сравнения контролируются на предмет на абсолютного, а относительно равенства.

Синтаксис

Коррекция точности при ошибках сравнения

TRUNC (R1*1000)

Значение

TRUNC:	Обрезание мест после запятой
--------	------------------------------

Учетное относительное равенство 10^{-12} при операциях

- Равенство: (==)
- Неравенство: (<>)
- Больше или равно: (>=)
- Меньше или равно: (<=)
- Больше/Меньше: (><) с абсолютным равенством
- Больше: (>)
- Меньше: (<)

Совместимость

По соображениям совместимости проверка на предмет относительного равенства при (>) и (<) может быть деактивирована через установку машинных данных MD10280 \$MN_PROG_FUNCTION_MASK Бит0 = 1.

Примечание

Поэтому сравнения с данными типа REAL по названным причинам всегда содержат определенную неточность. В случае неприемлемых погрешностей необходимо перейти на вычисление INTEGER, при этом операнды умножаются на коэффициент точности и после обрезаются с TRUNC.

Синхронные действия

Описанное поведение команд сравнения распространяется и на синхронные действия.

Примеры**Пример 1: Анализ точности**

Программный код	Комментарий
N40 R1=61.01 R2=61.02 R3=0.01	; присваивание начальных значений
N41 IF ABS(R2-R1) > R3 GOTOF FEHLER	; прежде переход был бы выполнен
N42 M30	; конец программы
N43 FEHLER: SETAL(66000)	
R1=61.01 R2=61.02 R3=0.01	; присваивание начальных значений
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000)	; коррекция точности
R13=TRUNC(R3*1000)	
IF ABS(R12-R11) > R13 GOTOF FEHLER	; переход более не выполняется
M30	; конец программы
FEHLER: SETAL(66000)	

Пример 2: Образование и обработка коэффициента двух операндов

Программный код	Комментарий
R1=61.01 R2=61.02 R3=0.01	; присваивание начальных значений
IF ABS((R2-R1)/R3)-1 > 10EX-5 GOTOF FEHLER	; переход не выполняется
M30	; конец программы
FEHLER: SETAL(66000)	

2.6 Минимум, максимум и диапазон переменных (MINVAL, MAXVAL и BOUND)

С помощью команд MINVAL и MAXVAL можно сравнить между собой значения двух переменных. Как результат возвращается меньшее значение (при MINVAL) или большее значение (при MAXVAL).

С помощью функции BOUND можно проверить, лежит ли значение контрольной переменной в пределах определенного диапазона значений.

Синтаксис

<меньшее значение>=MINVAL (<переменная1>,<переменная2>)
 <большее значение>=MAXVAL (<переменная1>,<переменная2>)
 <возвращаемое значение>=<BOUND> (<минимум>,<максимум>,<контрольная переменная>)

Значение

MINVAL:	Определяет меньшее значение двух переменных (<переменная1>,<переменная2>)
<меньшее значение>:	Переменная результата для команды MINVAL Устанавливается на меньшее значение переменной.
MAXVAL:	Определяет большее значение двух переменных (<переменная1>,<переменная2>)
<большее значение>:	Переменная результата для команды MAXVAL Устанавливается на большее значение переменной.
BOUND:	Проверяет, лежит ли переменная (<контрольная переменная>) в пределах определенного диапазона значений.
<минимум>:	Переменная, определяющая мин. значение диапазона значений
<максимум>:	Переменная, определяющая макс. значение диапазона значений
<возвращаемое значение>:	Переменная результата для команды BOUND Если значение контрольной переменной лежит в пределах определенного диапазона значений, то переменная результата устанавливается на значение контрольной переменной. Если значение контрольной переменной больше макс. значения, то переменная результата устанавливается на макс. значение области определения. Если значение контрольной переменной меньше мин. значения, то переменная результата устанавливается на мин. значение области определения.

Примечание

MINVAL, MAXVAL и BOUND могут программироваться и в синхронных действиях.

Примечание**Поведение при равенстве**

При равенстве для MINVAL/MAXVAL выводится это равное значение. При BOUND снова возвращается значение проверяемой переменной.

Пример

Программный код	Комментарий
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar	
rValMin=MINVAL(rVar1,rVar2)	; rValMin устанавливается на значение 10.5.
rValMax=MAXVAL(rVar1,rVar2)	; rValMax устанавливается на значение 33.7.
rVar3=19.7	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 лежит в пределах границ, rRetVar устанавливается на 19.7.
rVar3=1.8	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 лежит ниже мин. границы, rRetVar устанавливается на 10.5.
rVar3=45.2	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 лежит выше макс. границы, rRetVar устанавливается на 33.7.

2.7 Приоритет операций

Каждому оператору соответствует приоритет. При обработке выражения всегда сначала используются операторы с более высоким приоритетом. В случае равнозначных операторов обработка выполняется слева направо.

В арифметических выражениях посредством круглых скобок можно устанавливать последовательность выполнения всех операторов, отличную от обычных правил приоритета.

Последовательность операторов

От высшего к низшему приоритету

1.	NOT, B_NOT	Отрицание, побитовое отрицание
2.	*, /, DIV, MOD	Умножение, деление
3.	+, -	Сложение, вычитание
4.	B_AND	Побитовое И
5.	B_XOR	Побитовое исключающее ИЛИ
6.	B_OR	Побитовое ИЛИ
7.	AND	И
8.	XOR	Исключающее ИЛИ
9.	OR	ИЛИ
10.	<<	Соединение строк, тип результата STRING
11.	==, <>, >, <, >=, <=	Операторы сравнения

Примечание

Связывающий оператор ":" для фреймов не может использоваться с другими операторами в одном выражении. Поэтому градация приоритетов для этого оператора не требуется.

Пример оператора If

```
If (otto==10) and (anna==20) goto f end
```

2.8 Возможные преобразования типов

Функция

Преобразование типа при присваивании

Постоянное числовое значение, переменная или выражение, присваиваемое переменной, должно быть совместимо с типом этой переменной. Если это так, то при присваивании тип преобразуется автоматически.

Возможные преобразования типов

к позиции	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
из							
REAL	да	да*	да ¹⁾	да*	–	–	–
INT	да	да	да ¹⁾	да ²⁾	–	–	–
BOOL	да	да	да	да	да	–	–
CHAR	да	да	да ¹⁾	да	да	–	–
STRING	–	–	да ⁴⁾	да ³⁾	да	–	–
AXIS	–	–	–	–	–	да	–
FRAME	–	–	–	–	–	–	да

Пояснения

- * При преобразовании типа из REAL в INT дробное значение $\geq 0,5$ округляется в большую сторону, иные в меньшую сторону (см. функцию ROUND)
- ¹⁾ Значение $\neq 0$ соответствует TRUE, значение $= 0$ соответствует FALSE
- ²⁾ Если значение находится в допустимом числовом диапазоне
- ³⁾ Если только 1 символ
- ⁴⁾ Длина строки 0 = >FALSE, в иных случаях TRUE

Примечание

Если при преобразовании значение превышает конечный диапазон, то следует сообщение об ошибке.

Если в одном выражении встречаются различные типы, то согласование типов выполняется автоматически. Преобразования типов возможны и в синхронных действиях, см. главу "Синхронные действия движения, не явное преобразование типов".

2.9 Операции со строкой

Операции со строкой

Наряду с классическими операциями "Присваивание значения" и "Сравнение" возможны следующие операции со строкой:

- Преобразование типов в STRING (AXSTRING) (Страница 91)
- Преобразование типов из STRING (NUMBER, ISNUMBER, AXNAME) (Страница 92)
- Соединение строк (<<) (Страница 93)
- Преобразование в строчные/прописные буквы (TOLOWER, TOUPPER) (Страница 94)
- Определение длины строки (STRLEN) (Страница 95)
- Поиск символа/строки в строке (INDEX, RINDEX, MINDEX, MATCH) (Страница 96)
- Выбор части строки (SUBSTR) (Страница 97)
- Чтение и запись отдельных символов (Страница 98)
- Форматирование строки (SPRINT) (Страница 99)

Специальное значение символа 0

Символ 0 внутренне интерпретируется на идентификатор конца строки. При замене символа на символ 0 строка сокращается.

Пример:

Программный код	Комментарий
DEF STRING[20] STRG="Ось . остановлена"	
STRG[6]="X"	
MSG(STRG)	; Выводит сообщение "Ось x остановлена".
STRG[6]=0	
MSG(STRG)	; Выводит сообщение "Ось".

2.9.1 Преобразование типов в STRING (AXSTRING)

Благодаря функции "Преобразование типа в STRING" возможно использование переменных различного типа в качестве составной части сообщения (MSG).

Выполняется не явно при использовании оператора << для типов данных INT, REAL, CHAR и BOOL (см. "Соединение строк (<<) (Страница 93)").

Значение INT преобразуется в обычную читабельную форму. Для значений REAL указывается до 10 мест после запятой.

С помощью команды AXSTRING возможно преобразование переменных типа AXIS в STRING.

Синтаксис

```
<STRING_ERG> = << <bel._Тип>
<STRING_ERG> = AXSTRING (<идентификатор оси>)
```

Значение

<STRING_ERG>:	Переменная для результата преобразования типа	
	Тип:	STRING
<люб._тип>:	Типы переменных INT, REAL, CHAR, STRING и BOOL	
AXSTRING:	Команда AXSTRING выводит указанный идентификатор оси как строку.	
<идентификатор оси>:	Переменная для идентификатора оси	
	Тип:	AXIS

Примечание

Переменные FRAME не могут быть преобразованы.

2.9.2 Преобразование типов из STRING (NUMBER, ISNUMBER, AXNAME)

С помощью команды NUMBER выполняется преобразование из STRING в REAL. Возможность преобразования можно проверить с помощью команды ISNUMBER .

С помощью команды AXNAME строка преобразуется в тип данных AXIS.

Синтаксис

```
<REAL_ERG>=NUMBER ("<строка>")
<BOOL_ERG>=ISNUMBER ("<строка>")
<AXIS_ERG>=AXNAME ("<строка>")
```

Значение

NUMBER:	Команда NUMBER возвращает представленное через <строка> число как значение REAL.	
<строка>:	Преобразуемая переменная типа STRING	
<REAL_ERG>:	Переменная для результата преобразования типа с NUMBER	
	Тип:	REAL
ISNUMBER:	С помощью команды ISNUMBER можно проверить, может ли <строка> быть преобразован в действительное число.	

<BOOL_ERG>:	Переменная для результата опроса с ISNUMBER		
	Тип:	BOOL	
	Значение:	TRUE	ISNUMBER выводит значение TRUE, если <строка> представляет действительное по правилам языка число REAL.
		false	Если ISNUMBER выводит значение FALSE, то при вызове NUMBER с той же <строка> следует ошибка.
AXNAME:	Команда AXNAME преобразует указанную <строку> в идентификатор оси. Примечание: Если <строка> не может быть согласована с сконфигурированным идентификатором оси, то следует ошибка.		
<AXIS_ERG>:	Переменная для результата преобразования типа с AXNAME		
	Тип:	AXIS	

Пример

Программный код	Комментарий
DEF BOOL BOOL_ERG	
DEF REAL REAL_ERG	
DEF AXIS AXIS_ERG	
BOOL_ERG=ISNUMBER("1234.9876Eх-7")	; BOOL_ERG == TRUE
BOOL_ERG=ISNUMBER("1234XYZ")	; BOOL_ERG == FALSE
REAL_ERG=NUMBER("1234.9876Eх-7")	; REAL_ERG == 1234.9876Eх-7
AXIS_ERG=AXNAME("X")	; AXIS_ERG == X

2.9.3 Соединение строк (<<)

Функция "Соединение строк" позволяет составить строку из отдельных частей.

Связь реализуется через оператор "<<". Конечным типом этого оператора для всех комбинаций основных типов CHAR, BOOL, INT, REAL и STRING является STRING. Возможно необходимое преобразование выполняется по существующим правилам.

Синтаксис

<люб._тип> << <люб._тип>

Объяснение

<люб._тип>:	Переменные типа CHAR, BOOL, INT, REAL или STRING
<< :	Оператор для соединения переменных (<люб._тип>) в составную строку символов (тип STRING). Этот оператор доступен отдельно в качестве так называемого "унарного" варианта. Таким образом возможно явное преобразование типов в STRING (не для FRAME и AXIS): << <люб._тип>

К примеру, таким образом можно составить сообщение или команду из текстовых списков и вставить параметры (около одного имени блока):

```
MSG (STRG_TAB[LOAD_IDX]<<ИМЯ БЛОКА)
```

Примечание

Промежуточные результаты при соединении строк не могут превышать макс. длины строки.

Примечание

Типы FRAME и AXIS не могут использоваться с оператором "<<".

Примеры

Пример 1: Соединение строк

Программный код	Комментарий
DEF INT IDX=2	
DEF REAL VALUE=9.654	
DEF STRING[20] STRG="INDEX:2"	
IF STRG=="Индекс: "<<IDX GOTOF NO_MSG	
MSG ("Индекс: "<<IDX<<" / значение: "<<VALUE)	; Индикация: "Индекс:2/значение:9.654"
NO_MSG:	

Пример 2: Явное преобразование типа с <<

Программный код	Комментарий
DEF REAL VALUE=3.5	
<<VALUE	; Указанная переменная типа REAL преобразуется в тип STRING.

2.9.4 Преобразование в строчные/прописные буквы (TOLOWER, TOUPPER)

Функция "Преобразование в строчные/прописные буквы" позволяет преобразовать все буквы строки символов в унифицированное представление.

Синтаксис

```
<STRING_ERG>=TOUPPER ("<строка>")
<STRING_ERG>=TOLOWER ("<строка>")
```

Значение

TOUPPER:	С помощью команды TOUPPER все буквы строки символов преобразуются в прописные буквы.	
TOLOWER:	С помощью команды TOLOWER все буквы строки символов преобразуются в строчные буквы.	
<строка>:	Строка символов, которая должна быть преобразована	
	Тип:	STRING
<STRING_ERG>:	Переменная для результата преобразования	
	Тип:	STRING

Пример

Так как также существует возможность запуска данных пользователя на интерфейсе, то может быть достигнуто унифицированное представление со строчными или прописными буквами:

Программный код

```
DEF STRING [29] STRG
...
IF "LEARN.CNC"==TOUPPER(STRG) GOTOF LOAD_LEARN
```

2.9.5 Определение длины строки (STRLEN)

С помощью команды STRLEN можно определить длину строки символов.

Синтаксис

```
<INT_ERG>=STRLEN ("<СТРОКА>")
```

Значение

STRLEN:	С помощью команды STRLEN определяется длина указанной строки символов. Возвращается число символов, которые – считая от начала строки символов - не являются символами 0.	
<строка>:	Строка символов, длина которой должна быть определена	
	Тип:	STRING
<INT_ERG>:	Переменная для результата определения	
	Тип:	INT

Пример

Функция в комбинации с доступом к отдельным символам позволяет определить конец строки символов:

Программный код

```
IF (STRLEN(BAUSTEIN_NAME)>10) GOTO FFEHLER
```

2.9.6 Поиск символа/строки в строке (INDEX, RINDEX, MINDEX, MATCH)

Эта функция позволяет выполнять поиск отдельных символов или строки в другой строке. Результат функции показывает, на какой позиции строки был найден символ/строка в рассматриваемой строке.

Синтаксис

INT_ERG=INDEX (STRING, CHAR) ; Тип результата: INT

INT_ERG=RINDEX (STRING, CHAR) ; Тип результата: INT

INT_ERG=MINDEX (STRING, STRING) ; Тип результата: INT

INT_ERG=MATCH (STRING, STRING) ; Тип результата: INT

Семантика

Функции поиска: Они возвращают позицию в строке (первый параметр), где поиск был успешным. Если символ/строка не найдены, то возвращается значение -1. При этом первый символ имеет позицию 0.

Значение

INDEX:	Поиск указанного вторым параметром символа (сначала) в первом параметре.
RINDEX:	Поиск указанного вторым параметром символа (с конца) в первом параметре.
MINDEX:	Соответствует функции INDEX, за исключением того, что передается список символов (как строка), из которых возвращается индекс первого найденного символа.
MATCH:	Поиск строки в строке.

Таким образом, строки могут разбиваться по определенным критериям, к примеру, по позициям с пробелом или разделительному символу пути ("/").

Пример**Разделение введенных данных по имени пути и блока**

Программный код	Комментарий
DEF INT PFADIDX, PROGIDX	
DEF STRING[26] EINGABE	
DEF INT LISTIDX	

Программный код	Комментарий
EINGABE = "/_N_MPF_DIR/ _N_EXECUTE_MPF"	
LISTIDX = MINDEX (EINGABE, "M,N,O,P") + 1	; Как значение в LISTIDX возвращается 3; т.к. "N" это первый символ в параметре EINGABE, из списка выбора сначала.
PFADIDX = INDEX (EINGABE, "/") +1	; при этом: PFADIDX = 1
PROGIDX = RINDEX (EINGABE, "/") +1	; при этом: PROGIDX = 12
	; С помощью введенной в следующем разделе функции SUBSTR
VARIABLE = SUBSTR (EINGABE, PFADIDX, PROGIDX-PFADIDX-1)	переменная EINGABE может быть разбита на компонен- ты ;"Путь";и "Блок":
	; выводит "_N_MPF_DIR"
VARIABLE = SUBSTR (EINGABE, PROGIDX)	; выводит "_N_EXECUTE_MPF"

2.9.7 Выбор части строки (SUBSTR)

С помощью функции SUBSTRING возможно чтение любых частей в рамках строки.

Синтаксис

<STRING_ERG>=SUBSTR (<строка>,<индекс>,<длина>)

<STRING_ERG>=SUBSTR (<строка>,<индекс>)

Значение

SUBSTR:	Функция выводит из <строка> часть строки, исходя из <индекс> с указанной <длина>. Если параметр <длина> не указан, то функция выводит часть строки, исходя из <индекс> до конца строки.
<индекс>:	Начальная позиция части строки в рамках строки. Если начальная позиция лежит за концом строки, то возвращается пустая строка (" "). Первый символ строки: Индекс = 0 Диапазон значений: 0 ... (длина строки - 1)
<длина>:	Длина части строки. Если указывается слишком большая длина, то возвращается только часть строки до конца строки. Диапазон значений: 1 ... (длина строки - 1)

Пример

Программный код	Комментарий
DEF STRING[29] ERG	

Программный код	Комментарий
; 1	
; 0123456789012345678	
ERG = SUBSTR("QUITUNG: 10 до 99", 10, 2)	; ERG == "10"
ERG = SUBSTR("QUITUNG: 10 до 99", 10)	; ERG == "10 до 99"

2.9.8 Чтение и запись отдельных символов

В рамках строки возможно чтение и запись отдельных символов.

При этом необходимо соблюдать следующие граничные условия:

- возможно только для определенных пользователем переменных, не для системных переменных
- отдельные символы строки при вызовах подпрограмм передаются только "call by value"

Синтаксис

```
<символ>=<строка>[<индекс>]
<символ>=<строка_массив>[<массив_индекс>,<индекс>]
<строка>[<индекс>]=<символ>
<строка_массив>[<массив_индекс>,<индекс>]=<символ>
```

Значение

<строка>:	Любая строка
<символ>:	Переменная типа CHAR
<индекс>:	Позиция символа в рамках строки. Первый символ строки: Индекс = 0 Диапазон значений: 0 ... (длина строки - 1)

Примеры

Пример 1: Переменное сообщение

Программный код	Комментарий
; 0123456789	
DEF STRING [50] MELDUNG = "Ось n достигла позиции"	
MELDUNG [6] = "x"	
MSG (MELDUNG)	; "Ось x достигла позиции"

Пример 2: Обработка системной переменной

Программный код	Комментарий
DEF STRING [50] STRG	; Буфер для системных переменных
...	

Программный код	Комментарий
STRG = \$P_MMCA	; Загрузка системных переменных
IF STRG[0] == "E" GOTO ...	; Обработка системных переменных

Пример 3: Передача параметров "вызов по значению" и "вызов по ссылке"

Программный код	Комментарий
; 0123456	
DEF STRING[50] STRG = "ось X"	
DEF CHAR CHR	
...	
EXTERN UP_VAL(ACHSE)	; Определение подпрограммы с параметром "вызов по значению"
EXTERN UP_REF(VAR ACHSE)	; Определение подпрограммы с параметром "вызов по ссылке"
...	
UP_VAL(STRG[6])	; Передача параметров "по значению"
...	
CHR = STRG[6]	; Буферизация
UP_REF(CHR)	; Передача параметров "по ссылке"

2.9.9 Форматирование строки (SPRINT)

С помощью предопределенной функции SPRINT можно форматировать и, к примеру, подготавливать строки символов для вывода на внешние устройства (см. также "Process DataShare - Вывод на внешнее устройство/файл (EXTOPEN, WRITE, EXTCLOSE) (Страница 700)").

Синтаксис

"<результат_строка>"=SPRINT ("<формат_строка>",<значение_1>,<значение_2>,...,<значение_n>)

Значение

SPRINT:	Идентификатор для предопределенной функции, выводящей значение типа STRING.
"<формат_строка>":	Строка символов, содержащая постоянные и переменные части. Переменные части определяются знаком управления форматом % и последующим описанием формата.
< значение_1>,< значение_2>,...,< значение_n>:	Значение в форме постоянной или переменной ЧПУ, вставляемое в месте, в котором стоит n-ый знак управления форматом %, согласно описанию формата в <формат_строка>.
"<результат_строка>":	Форматированная строка символов (макс. 400 байт)

Имеющиеся описания форматов

%B:	Преобразование в строку "TRUE", если преобразуемое значение: - отлично от 0. - не является пустой строкой (для значений String). Преобразование в строку "FALSE", если преобразуемое значение: - равно 0. - является пустой строкой. Пример: <pre>N10 DEF BOOL BOOL_VAR=1 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF BOOL_VAR:TRUE".
%C:	Преобразование с символ ASCII. Пример: <pre>N10 DEF CHAR CHAR_VAR="X" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C", CHAR_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF CHAR_VAR:X".
%D:	Преобразование в строку с целочисленным значением (INTEGER). Пример: <pre>N10 DEF INT INT_VAR=123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D", INT_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF INT_VAR:123".
%<m>D:	Преобразование в строку с целочисленным значением (INTEGER). Строка имеет мин. длину в <m> символов. Отсутствующие места заполняются выровненными по левым разрядам символами пробела. Пример: <pre>N10 DEF INT INT_VAR=-123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D", INT_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF INT_VAR:xx-123" ("x" в примере замещает символ пробела).
%F:	Преобразование в строку с десятичным числом с 6 местами после запятой. Места после запятой при необходимости округляются или заполняются 0. Пример: <pre>N10 DEF REAL REAL_VAR=-1.2341234EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F", REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:-1234.123400".

%<m>F:	Преобразование в строку с десятичным числом с 6 местами после запятой и общей длиной мин. в <m> символов. Места после запятой при необходимости округляются или заполняются 0. Отсутствующие символы на всю длину <m> заполняются выровненными по левым разрядам символами пробела. Пример: <pre>N10 DEF REAL REAL_VAR=-1.23412345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR: xxx-1234.123457" ("x" в примере замещает символ пробела).
%.<n>F:	Преобразование в строку с десятичным числом с <n> местами после запятой. Места после запятой при необходимости округляются или заполняются 0. Пример: <pre>N10 DEF REAL REAL_VAR=-1.2345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:-1234.568".
%<m>.<n>F:	Преобразование в строку с десятичным числом с <n> местами после запятой и общей длиной мин. в <m> символов. Места после запятой при необходимости округляются или заполняются 0. Отсутствующие символы на всю длину <m> заполняются выровненными по левым разрядам символами пробела. Пример: <pre>N10 DEF REAL REAL_VAR=-1.2341234567890EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:xx-1234.12" ("x" в примере замещает символ пробела).
%E:	Преобразование в строку с десятичным числом в экспоненциальном представлении. Мантисса сохраняется нормализованной с одним местом перед запятой и 6 местами после запятой. Места после запятой при необходимости округляются или заполняются 0. Показатель начинается с кодового слова "EX". За ним следует знак ("+" или "-") и двух- или трехзначное число. Пример: <pre>N10 DEF REAL REAL_VAR=-1234.567890 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:-1.234568EX+03".

%<m>E:	Преобразование в строку с десятичным числом в экспоненциальном представлении с общей длиной мин. в <m> символов. Отсутствующие символы заполняются выровненными по левым разрядам символами пробела. Мантисса сохраняется нормализованной с одним местом перед запятой и 6 местами после запятой. Места после запятой при необходимости округляются или заполняются 0. Показатель начинается с кодового слова "EX". За ним следует знак ("+" или "-") и двух- или трехзначное число. Пример: <pre>N10 DEF REAL REAL_VAR=-1234,5 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03" ("x" в примере замещает символ пробела)
%.<n>E:	Преобразование в строку с десятичным числом в экспоненциальном представлении. Мантисса сохраняется нормализованной с одним местом перед запятой и <n> местами после запятой. Места после запятой при необходимости округляются или заполняются 0. Показатель начинается с кодового слова "EX". За ним следует знак ("+" или "-") и двух- или трехзначное число. Пример: <pre>N10 DEF REAL REAL_VAR=-1234,5678 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:-1.23EX+03".
%<m>.<n>E:	Преобразование в строку с десятичным числом в экспоненциальном представлении с общей длиной мин. в <m> символов. Отсутствующие символы заполняются выровненными по левым разрядам символами пробела. Мантисса сохраняется нормализованной с одним местом перед запятой и <n> местами после запятой. Места после запятой при необходимости округляются или заполняются 0. Показатель начинается с кодового слова "EX". За ним следует знак ("+" или "-") и двух- или трехзначное число. Пример: <pre>N10 DEF REAL REAL_VAR=-1234,5678 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:xx-1.23EX+03" ("x" в примере замещает символ пробела).

%G:	Преобразование в строку с десятичным числом в десятичном или экспоненциальном представлении в зависимости от диапазона значений: если представляемое значение по величине меньше 1.0EX-04 или больше/равно 1.0EX+06, то выбирается экспоненциальное представление, в остальных случаях десятичное представление. Отображается макс. шесть значимых мест, при необходимости выполняется округление. Пример с десятичным представлением: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:0,000123457". Пример с экспоненциальным представлением: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+06 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:1.23457EX+06".
%<m>G:	Преобразование в строку с десятичным числом в десятичном или экспоненциальном представлении в зависимости от диапазона значений (как %G). Строка имеет общую длину мин. в <m> символов. Отсутствующие символы заполняются выровненными по левым разрядам символами пробела. Пример с десятичным представлением: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:xxxx0.000123457" ("x" в примере замещает символ пробела). Пример с экспоненциальным представлением: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+06 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:xxx1.23457EX+06" ("x" в примере замещает символ пробела).

%.<n>G:	Преобразование в строку с десятичным числом в десятичном или экспоненциальном представлении в зависимости от диапазона значений. Отображается макс. <n> значимых мест, при необходимости выполняется округление. если представляемое значение по величине меньше 1.0EX-04 или больше/равно 1.0EX(+<n>), то выбирается экспоненциальное представление, в остальных случаях десятичное представление. Пример с десятичным представлением: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:0,000123". Пример с экспоненциальным представлением: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+03 N20 DEF STRING[80] RESULT N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:1.23EX+03".
%<m>.<n>G:	Преобразование в строку с десятичным числом в десятичном или экспоненциальном представлении в зависимости от диапазона значений (как %.<n>G). Строка имеет общую длину мин. в <m> символов. Отсутствующие символы заполняются выровненными по левым разрядам символами пробела. Пример с десятичным представлением: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:xxx0.0001235" ("x" в примере замещает символ пробела). Пример с экспоненциальным представлением: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF REAL_VAR:xx1.235EX+06" ("x" в примере замещает символ пробела).

%.<n>P:	Преобразование REAL-значения в INTEGER-значение с учетом <n> мест после запятой. INTEGER-значение выводится как 32-битное двоичное число. Если преобразуемое значение не может быть представлено в 32 бит, то обработка отменяется с ошибкой. Т.к. созданная с оператором спецификации формата %.<n>P байтовая последовательность может содержать и двоичные нули, то созданная таким образом общая строка более не соответствует условным обозначениям типа данных ЧПУ STRING. Поэтому она не может быть ни сохранена в переменную типа STRING, ни подвергнута дальнейшей обработке с помощью строковых команд языка ЧПУ. Единственным возможным применением является передача параметров на команду WRITE с выводом на соответствующее внешнее устройство (см. пример ниже). Как только <формат_строка> содержит описание формата типа %P, вся строка, за исключением сгенерированного с %.<n>P двоичного числа, выводится согласно MD10750 \$MN_SPRINT_FORMAT_P_CODE в символьном коде ASCII, ISO (DIN6024) или EIA (RS244). При программировании символа, который не может быть преобразован, обработка отменяется с ошибкой. Пример: <pre> N10 DEF REAL REAL_VAR=123.45 N20 DEF INT ERROR N30 DEF STRING[20] EXT_DEVICE="/ext/dev/1" ... N100 EXTOPEN(ERROR,EXT_DEVICE) N110 IF ERROR <> 0 ... ; обработка ошибок N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%. 3P",REAL_VAR) N210 IF ERROR <> 0 ... ; обработка ошибок </pre> Результат: Строка "INTEGER BINARY CODED: 'H0001E23A'" передается на устройство вывода /ext/dev/1. Шестнадцатеричное значение 0x0001E23A соответствует десятичному значению 123450.
---------	--

%<m>.<n>P:	Преобразование REAL-значения согласно установке в машинных данных MD10751 \$MN_SPRINT_FORMAT_P_DECIMAL в строку с: • целым числом из <m> + <n> мест или • десятичным числом макс. с <m> местами перед запятой и точно <n> местами после запятой. Как у описания формата %.<n>P вся строка сохраняется в определенный через MD10750 \$MN_SPRINT_FORMAT_P_CODE код символа. Преобразование при MD10751 = 0: Значение REAL преобразуется в строку с целым числом из <m> + <n> мест. При необходимости места после запятой округляются до <n> мест или заполняются нулями. Отсутствующие места перед запятой заполняются символами пробела. Знак минус добавляется выровненным слева, вместо знака плюс вставляется пробел. Пример: <pre>N10 DEF REAL REAL_VAR=-123.45 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "PUNCHED TAPE FORMAT:-xx123450" ("x" в примере замещает символ пробела). Преобразование при MD10751 = 1: Значение REAL преобразуется в строку с десятичным числом макс. <m> местами перед запятой и точно <n> местами после запятой. При необходимости места перед запятой обрезаются, а места после запятой округляются или заполняются нулями. Если <n> равно 0, то десятичная точка также не нужна. Пример: <pre>N10 DEF REAL REAL_VAR1=-123.45 N20 DEF REAL REAL_VAR2=123.45 N30 DEF STRING[80] RESULT N40 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P VAR2:%2.0P", REAL_VAR1,REAL_VAR2)</pre> Результат: Строковая переменная RESULT описывается строкой символов "PUNCHED TAPE FORMAT:-123.450 VAR2:23".
%S:	Вставка строки. Пример: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF STRING_VAR:ABCDEFGH".
%<m>S:	Вставка строки мин. с <m> символами. Отсутствующие места заполняются символами пробела. Пример: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF STRING_VAR:xxxABCDEFGH" ("x" в примере замещает символ пробела).

<code>%.<n>S:</code>	Вставка <n> символов строки (начиная с первого символа). Пример: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S", STRING_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF STRING_VAR:ABC".
<code>%<m>.<n>S:</code>	Вставка <n> символов строки (начиная с первого символа). Созданная строка имеет общую длину мин. в <m> символов. Отсутствующие места заполняются символами пробела. Пример: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "CONTENT OF STRING_VAR:xxxxxABCDE" ("x" в примере замещает символ пробела).
<code>%X:</code>	Преобразование значения INTEGER в строку с шестнадцатеричным представлением. Пример: <pre>N10 DEF INT INT_VAR='HA5B8' N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X", INT_VAR)</pre> Результат: Строковая переменная RESULT описывается строкой символов "INTEGER HEXADECIMAL:A5B8".

Примечание

Свойство языка ЧПУ, не делающего различий между прописным и строчным написанием идентификаторов и кодовых слов, действует и для описаний форматов. Поэтому они могут программироваться прописными или строчными буквами без различия в функциях.

Возможности образования комбинаций

Таблица ниже показывает, какие типы данных ЧПУ могут комбинироваться с каким описанием формата. Действуют правила для не явного преобразования типа данных (см. "Типы данных (Страница 63)").

	Типы данных ЧПУ						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-

	Типы данных ЧПУ						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%X	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

Примечание

Из таблицы видно, что типы данных ЧПУ AXIS и FRAME не могут использоваться напрямую в функции SPRINT. Но можно:

- преобразовать тип данных AXIS с помощью функции AXSTRING в строку, которая после может быть обработана со SPRINT.
 - прочитать отдельные значения типа данных FRAME через доступ к фрейм-компонентам. Благодаря этому получаются данные типа REAL, которые могут быть подвергнуты дальнейшей обработке со SPRINT.
-

2.10 Переходы и ветвления в программе

2.10.1 Возврат на начало программы (GOTOS)

С помощью команды `GOTOS` для повторения программы можно вернуться на начало главной или подпрограммы.

Через машинные данные можно установить, чтобы при каждом возврате на начало программы:

- время выполнения программы устанавливалось бы на "0".
- подсчет деталей увеличивался бы на значение "1".

Синтаксис

`GOTOS`

Значение

GOTOS:	Оператор перехода с целью перехода "начало программы".	
	Выполнение управляется через интерфейсный сигнал ЧПУ/PLC: DB21, ... DBX384.0 (управление ветвлением программы)	
	Величина:	Значение:
	0	Нет возврата на начало программы. Выполнение программы продолжается на следующем после <code>GOTOS</code> кадре программы обработки детали.
	1	Возврат на начало программы. Программа обработки детали повторяется.

Граничные условия

- `GOTOS` вызывает внутреннюю `STOPRE` (остановка предварительной обработки).
- В программе обработки детали с определениями данных (переменные `LUD`) с `GOTOS` выполняется переход на первый кадр программы после раздела определения, т. е. определения данных не осуществляются повторно. Таким образом, определенные переменные сохраняют достигнутое в кадре `GOTOS` значение и не сбрасываются на запрограммированные в разделе определения стандартные значения.
- В синхронных действиях и технологических циклах команда `GOTOS` недоступна.

Пример

Программный код	Комментарий
N10 ...	; Начало программы.
...	

Программный код	Комментарий
N90 GOTOS	; Переход на начало программы.
...	

2.10.2 Переходы на метки перехода (GOTOB, GOTOF, GOTO, GOTOS)

В программе могут быть установлены метки перехода (метки), на которые можно переходить из других мест внутри одной программы с помощью команд GOTOF, GOTOB, GOTO или GOTOS. После выполнения программы начинается на операторе, который следует непосредственно за меткой перехода. Благодаря этому могут быть реализованы ветвления внутри программы.

Наряду с метками перехода в качестве целей перехода возможны и номера главных и вспомогательных кадров.

Если перед оператором перехода сформулировано условие перехода (IF ...), то переход в программе осуществляется только при выполнении условия перехода.

Синтаксис

GOTOB <цель перехода>
IF <условие перехода> == TRUE GOTOB <цель перехода>

GOTOF <цель перехода>
IF <условие перехода> == TRUE GOTOF <цель перехода>

GOTO <цель перехода>
IF <условие перехода> == TRUE GOTO <цель перехода>

GOTOS <цель перехода>
IF <условие перехода> == TRUE GOTOS <цель перехода>

Значение

GOTOB:	Оператор перехода с целью перехода в направлении начала программы.
GOTOF:	Оператор перехода с целью перехода в направлении конца программы.
GOTO:	Оператор перехода с поиском цели перехода. Поиск сначала выполняется в направлении конца программы, потом в направлении начала программы.
GOTOS:	Действие аналогично GOTO с той разницей, что ошибка 14080 "Цель перехода не найдена" подавляется. Это означает, что выполнение программы в случае безрезультатного поиска цели перехода не отменяется, а продолжается со следующей за командой GOTOS строки программы.

<цель перехода>:	Параметры цели перехода Возможными данными являются:	
	<метка перехода>:	Цель перехода это установленная в программе метка перехода с пользовательским именем: <метка перехода>:
	<номер кадра>:	Цель перехода это номер главного или вспомогательного кадра (к примеру: 200, N300)
	Переменная типа STRING:	Переменная цель перехода. Переменная обозначает метку перехода или номер кадра.
IF:	Кодовое слово для формулирования условия перехода. Условие перехода допускает все операции сравнения и логические операции (результат: TRUE или FALSE). Переход в программе выполняется, если результат этой операции TRUE.	

Примечание**Метки перехода (метки)**

Метки перехода всегда стоят в начале кадра. Если имеется номер программы, то метка перехода стоит непосредственно за номером кадра.

Для наименования меток перехода действуют следующие правила:

- Число символов:
 - минимум 2
 - максимум 32
- Разрешенными символами являются:
 - буквы
 - цифры
 - символы подчеркивания
- Первыми двумя символами должны быть буквы или символы подчеркивания.
- После имени метки перехода следует двоеточие (":").

Граничные условия

- Целью перехода может быть только кадр с меткой перехода или номером кадра, лежащий **внутри** программы.
- Оператор перехода без условия перехода должен быть запрограммирован в отдельном кадре. Для операторов перехода с условиями перехода это ограничение не действует. Здесь можно сформулировать несколько операторов перехода в одном кадре.
- Для программ с операторами перехода без условий перехода конец программы M2/M30 не обязательно должен стоять на конце программы.

Примеры

Пример 1: Переходы на метки перехода

Программный код	Комментарий
N10 ...	
N20 GOTOF Label_1	; переход в направлении конца программы к метке перехода «Label_1».
N30 ...	
N40 Label_0: R1=R2+R3	; Метка перехода "Label_0" установлена.
N50 ...	
N60 Label_1:	; Метка перехода "Label_1" установлена.
N70 ...	
N80 GOTOB Label_0	; переход в направлении начала программы к метке перехода «Label_0».
N90 ...	

Пример 2: Косвенный переход на номер кадра

Программный код	Комментарий
IF <условие> == TRUE	
R10=100	; назначение цели перехода
ELSE	
R10=110	; назначение цели перехода
ENDIF	
; переход в направлении конца программы на кадр, номер которого стоит в R10	
N10 GOTOF "N"<<R10	
...	
N90 ...	
N100 ...	; Цель перехода
N110 ...	
...	

Пример 3: Переход на переменную цель перехода

Программный код	Комментарий
DEF STRING[20] ZIEL	
IF <условие> == TRUE	
ZIEL = "Marke1"	; назначение цели перехода
ELSE	
ZIEL = "Marke2"	; назначение цели перехода
ENDIF	
; переход в направлении конца программы к переменной цели перехода «Содержание ZIEL»	
GOTOF ZIEL	
Marke1: T="Bohrer1"	; цель перехода 1
...	
Marke2: T="Bohrer2"	; цель перехода 2

Программный код	Комментарий
...	

Пример 4: Переход с условием перехода

Программный код	Комментарий
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	; присваивание начальных значений
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6	; метка перехода LA1
N42 R1=R1+R3 R4=R4-1	
; IF условие перехода == TRUE	
; THEN переход в направлении начала программы к метке перехода LA1	
N43 IF R4>0 GOTO LA1	
N44 M30	; конец программы

2.10.3 Ветвление программы (CASE ... OF ... DEFAULT ...)

Функция CASE предлагает возможность проверки актуального значения (тип: INT) переменной или функции вычисления и в зависимости от результата перейти на различные места в программе.

Синтаксис

```
CASE (<выражение>) OF <постоянная_1> GOTO <цель перехода_1>
<постоянная_2> GOTO <цель перехода_2> ... DEFAULT GOTO <цель
перехода_n>
```

Значение

CASE:	Оператор перехода	
<выражение>:	Переменная или функция вычисления	
OF:	Кодовое слово для формулировки условных ветвлений программы	
<постоянная_1>:	Первое указанное постоянное значение для переменной или функции вычисления	
	Тип:	INT
<постоянная_2>:	Второе указанное постоянное значение для переменной или функции вычисления	
	Тип:	INT
DEFAULT:	На тот случай, когда переменная или функция вычисления не принимает ни одного из указанных постоянных значений, с помощью оператора DEFAULT можно определить цель перехода. Примечание: Если оператор DEFAULT не запрограммирован, то в этом случае целью перехода становится следующий за оператором CASE кадр.	
GOTO:	Оператор перехода с целью перехода в направлении конца программы. Вместо GOTO могут быть запрограммированы и все другие команды GOTO (см. тему "Переходы в программе на метки перехода").	

<цель перехода_1>:	Выполняется ветвление на эту цель перехода, если значение переменной или функции вычисления соответствует первой указанной постоянной.	
	Цель перехода может быть указана следующим образом:	
	<метка перехода>:	Цель перехода это установленная в программе метка перехода с пользовательским именем: <метка перехода>:
	<номер кадра>:	Цель перехода это номер главного или вспомогательного кадра (к примеру: 200, N300)
<цель перехода_2>:	Переменная типа STRING:	Переменная цель перехода. Переменная обозначает метку перехода или номер кадра.
	Выполняется ветвление на эту цель перехода, если значение переменной или функции вычисления соответствует второй указанной постоянной.	
<цель перехода_n>:	Выполняется ветвление на эту цель перехода, если значение переменной не принимает ни одного из указанных постоянных значений.	

Пример

Программный код

```

...
N20 DEF INT VAR1 VAR2 VAR3
N30 CASE (VAR1+VAR2-VAR3) OF 7 GOTO Label_1 9 GOTO
Label_2 DEFAULT GOTO Label_3
N40 Label_1: G0 X1 Y1
N50 Label_2: G0 X2 Y2
N60 Label_3: G0 X3 Y3
...

```

Оператор CASE из N30 определяет следующие возможности ветвления программы:

1. Если значение функции вычисления $VAR1+VAR2-VAR3 = 7$, то перейти на кадр с определением метки перехода "Label_1" ($\rightarrow$ N40).
2. Если значение функции вычисления $VAR1+VAR2-VAR3 = 9$, то перейти на кадр с определением метки перехода "Label_2" ($\rightarrow$ N50).
3. Если значение функции вычисления $VAR1+VAR2-VAR3$ ни 7, ни 9, то перейти на кадр с определением метки перехода "Label_3" ($\rightarrow$ N60).

2.11 Повторение блока программы (REPEAT, REPEATB, ENDLABEL, P)

Повторение программного блока позволяет повторять уже записанные блоки программы внутри программы в любом составе.

Повторяемые строки или области программы обозначаются метками перехода (метками).

Примечание

Метки перехода (метки)

Метки перехода всегда стоят в начале кадра. Если имеется номер программы, то метка перехода стоит непосредственно за номером кадра.

Для наименования меток перехода действуют следующие правила:

- Число символов:
 - минимум 2
 - максимум 32
 - Разрешенными символами являются:
 - буквы
 - цифры
 - символы подчеркивания
 - Первыми двумя символами должны быть буквы или символы подчеркивания.
 - После имени метки перехода следует двоеточие (":").
-

Синтаксис

1. Повторение отдельной строки программы:

```
<метка перехода>: ...  
...  
REPEATB <метка перехода> P=<n>  
...
```

2. Повторение области программы между меткой перехода и оператором REPEAT:

```
<метка перехода>: ...  
...  
REPEAT <метка перехода> P=<n>  
...
```

3. Повторение области между двумя метками перехода:

```
<стартовая метка перехода>: ...  
...  
<конечная метка перехода>: ...  
...
```

```

REPEAT <стартовая метка перехода> <конечная метка перехода>
P=<n>
...

```

Примечание

Заключение оператора REPEAT с обеими метками перехода в скобки невозможно. Если <стартовая метка перехода> найдена перед оператором REPEAT и <конечная метка перехода> не достигается до оператора REPEAT, то выполняется повторение между <стартовая метка перехода> и оператором REPEAT.

4. Повторение области между меткой перехода и ENDLABEL:

```

<метка перехода>: ...
...
ENDLABEL: ...
...
REPEAT <метка перехода> P=<n>
...

```

Примечание

Заключение оператора REPEAT с <метка перехода> и ENDLABEL в скобки невозможно. Если <метка перехода> найдена перед оператором REPEAT и ENDLABEL не достигается до оператора REPEAT, то выполняется повторение между <метка перехода> и оператором REPEAT.

Значение

REPEATB:	Команда повторения строки программы
REPEAT:	Команда повторения области программы
<метка перехода>:	<метка перехода> обозначает: - повторяемую строку программы (при REPEATB) или - начало повторяемой области программы (при REPEAT) Обозначенная с <метка перехода> строка программы может стоять до или после оператора REPEAT/REPEATB. Поиск сначала выполняется в направлении начала программы. Если метка перехода в этом направлении не найдена, то выполняется поиск в направлении конца программы. Исключение: Если необходимо повторить область программы между меткой перехода и оператором REPEAT (см. 2. в Синтаксисе), то обозначенная с <метка перехода> строка программы должна стоять до оператора REPEAT, т.к. в этом случае поиск выполняется только в направлении начала программы. Если в кадре с <метка перехода> имеются дополнительные операторы, то при каждом повторении они выполняются заново.

2.11 Повторение блока программы (REPEAT, REPEATB, ENDLABEL, P)

ENDLABEL:	Кодовое слово, обозначающее конец повторяемой области программы Если строка с ENDLABEL содержит другие операторы, то при каждом повторении они выполняются заново. ENDLABEL может использоваться в программе многократно.	
P:	Адрес для указания числа повторов	
<n>:	Число повторов блока программы	
	Тип:	INT
	Повторяемый блок программы повторяется <n> раз. После последнего повторения программа продолжается на следующей за строкой REPEAT/REPEATB строке. Примечание: Если P=<n> не указано, то повторяемый блок программы повторяется точно один раз.	

Примеры

Пример 1: Повторение отдельной строки программы

Программный код	Комментарий
N10 POSITION1: X10 Y20	
N20 POSITION2: CYCLE(0,,9,8)	; Позиционный цикл
N30 ...	
N40 REPEATB POSITION1 P=5	; Выполнить КАДР N10 пять раз.
N50 REPEATB POSITION2	; Выполнить кадр N20 один раз.
N60 ...	
N70 M30	

Пример 2: Повторение области программы между меткой перехода и оператором REPEAT

Программный код	Комментарий
N5 R10=15	
N10 BEGIN: R10=R10+1	; Ширина
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	
N50 X=-R10	
N60 Y=-R10	
N70 Z=10+R10	
N80 REPEAT BEGIN P=4	; Выполнить область N10 до N70 четыре раза.
N90 Z10	
N100 M30	

Пример 3: Повторение области между двумя метками перехода

Программный код	Комментарий
N5 R10=15	

2.11 Повторение блока программы (REPEAT, REPEATB, ENDLABEL, P)

Программный код	Комментарий
N10 BEGIN: R10=R10+1	; Ширина
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	
N50 X=-R10	
N60 Y=-R10	
N70 END: Z=10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	; Выполнить область N10 до N70 три раза.
N110 Z10	
N120 M30	

Пример 4: Повторение области между меткой перехода и ENDLABEL

Программный код	Комментарий
N10 G1 F300 Z-10	
N20 BEGIN1:	
N30 X10	
N40 Y10	
N50 BEGIN2:	
N60 X20	
N70 Y30	
N80 ENDLABEL: Z10	
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3: X20	
N120 Y30	
N130 REPEAT BEGIN3 P=3	; Выполнить область N110 до N120 три раза.
N140 REPEAT BEGIN2 P=2	; Выполнить область N50 до N80 два раза.
N150 M100	
N160 REPEAT BEGIN1 P=2	; Выполнить область N20 до N80 два раза.
N170 Z10	
N180 X0 Y0	
N190 M30	

Пример 5: Фрезерная обработка, обработка позиции сверления с помощью различных технологий

Программный код	Комментарий
N10 ZENTRIERBOHRER()	; Установить центровое сверло.
N20 POS_1:	; Позиции сверления 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	

2.11 Повторение блока программы (REPEAT, REPEATB, ENDLABEL, P)

Программный код	Комментарий
N70 ENDLABEL:	
N80 POS_2:	; Позиции сверления 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL:	
N130 BOHRER()	; Сменить сверло и цикл сверления.
N140 GEWINDE(6)	; Установить метчик M6 и цикл резьбы.
N150 REPEAT POS_1	; Повторить сегмент программы от POS_1 один раз до ENDLABEL.
N160 BOHRER()	; Сменить сверло и цикл сверления.
N170 GEWINDE(8)	; Установить метчик M8 и цикл резьбы.
N180 REPEAT POS_2	; Повторить сегмент программы от POS_2 один раз до ENDLABEL.
N190 M30	

Дополнительная информация

- Возможен вложенный вызов повторения блока программы. Каждый вызов занимает один уровень подпрограммы.
- Если при обработке повторения блока программы программируется M17 или RET, то повторение блока программы отменяется. Программа продолжается на следующем за строкой REPEAT кадре.
- В актуальной индикации программы повторение блока программы индицируется как отдельный уровень подпрограммы.
- Если при обработке блока программы запускается разрыв уровней, то программа продолжается после вызова обработки блока программы.

Пример:

Программный код	Комментарий
N5 R10=15	
N10 BEGIN: R10=R10+1	; Ширина
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	; Разрыв уровней
N50 X=-R10	
N60 Y=-R10	
N70 END: Z10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	
N120 Z10	; Продолжить обработку программы.
N130 M30	

2.11 Повторение блока программы (REPEAT, REPEATB, ENDLABEL, P)

- Возможно комбинированное использование управляющих структур и повторения блока программы. Но не должно быть дублирований. Повторение блока программы должно лежать внутри ветви управляющей структуры или управляющая структура внутри повторения блока программы.
- При комбинации переходов и повторения блока программы кадры обрабатываются чисто последовательно. Если, к примеру, осуществляется переход из повторения блока программы, то обработка продолжается до тех пор, пока запрограммированный конец блока программы не будет найден.
Пример:

Программный код

```
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X=10
N40 Y=10
N50 GOTOF BEGIN2
N60 ENDLABEL:
N70 BEGIN2:
N80 X20
N90 Y30
N100 ENDLABEL: Z10
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30
```

Примечание

Оператор REPEAT должен стоять за кадрами перемещения.

2.12 Управляющие структуры

СЧПУ стандартно обрабатывает кадры ЧПУ в запрограммированной последовательности.

Эта последовательность может изменяться через программирование программных блоков и программных циклов. Эти управляющие структуры программируются с кодовыми словами IF, ELSE, ENDIF, LOOP, FOR, WHILE и REPEAT.

ВНИМАНИЕ

Ошибки программирования

Управляющие структуры возможны только внутри командной части программы. Условное или повторное выполнение определений в заголовке программы невозможно.

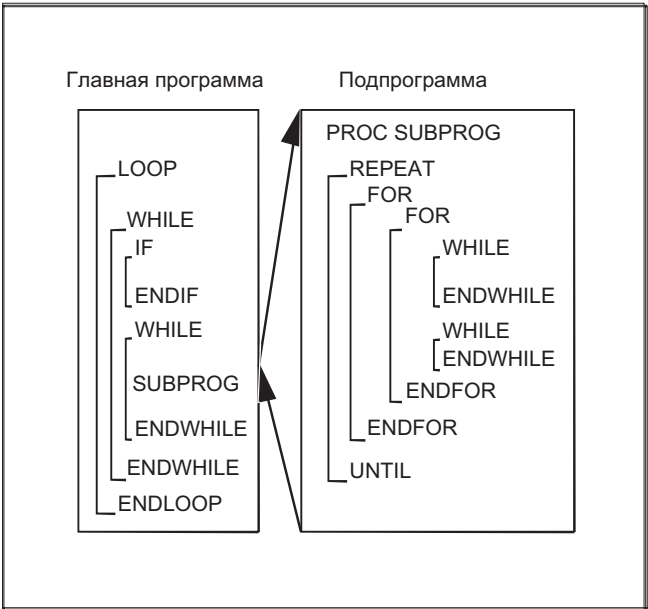
Кодовые слова для управляющих структур, как и цели перехода, не могут перекрываться макросами. Проверка для макроопределения не выполняется.

Активность

Управляющая структура не может использоваться между программами.

Глубина вложенности

Внутри каждого уровня подпрограммы возможна глубина вложенности до 16 управляющих структур.



Динамическая характеристика

В стандартно активном режиме интерпретатора посредством использования переходов в программе может быть достигнуто более быстрое выполнение программы, чем при использовании управляющих структур.

В предварительно скомпилированных циклах различие между переходами в программе и управляющими структурами отсутствует.

Текущая индикация кадров в программных циклах

Если в программном цикле выполняются только кадры предварительной обработки, то в текущей индикации кадров отображается последний кадр главного хода **перед** программным циклом.

Для отображения, например, с целью диагностики, и выполненных кадров предварительной обработки в текущей индикации кадров, необходимо активировать отдельный кадр декодирования SBL2.

Литература

Описание функций "Основные функции", глава: GPP, канал, программный режим, реакция на Reset (K1) > Отдельный кадр декодирования SBL2 с не явной остановкой предварительной обработки

Шлифование без кадра главного хода

Если в программном цикле не запрограммирован кадр главного хода, цикл обрабатывается в предварительной обработке, пока не выполнено условие цикла.

Тем самым возникает высокая загруженность, и индикация может оказаться под угрозой.

Способом устранения может быть добавление команды STOPRE или времени ожидания G04 в 0 секунд в цикл.

Граничные условия

- Пропуск кадров с элементами управляющих структур невозможен.
- Метки перехода (метки) не разрешены в кадрах с элементами управляющих структур.
- Управляющие структуры обрабатываются интерпретаторски. При определении конца цикла, с учетом найденных при этом управляющих структур, выполняется поиск начала цикла. Поэтому в режиме интерпретатора блочная структура программы не проверяется полностью.
- Не рекомендуется комбинированное использование управляющих структур и переходов в программе.
- При предварительной обработке циклов возможна проверка правильности вложенности управляющих структур.

2.12.1 Условный оператора и ветвление (IF, ELSE, ENDIF)

Условный оператор: IF - программный блок - ENDIF

В случае условного оператора стоящий между IF и ENDIF программный блок выполняется, только если условие выполнено.

Ветвление: IF - программный блок_1 - ELSE - программный блок_2 - ENDIF

В случае ветвления всегда выполняется один из двух программных блоков.

Стоящий между IF и ELSE программный блок_1 выполняется, если условие выполнено.

Выполняется стоящий между ELSE и ENDIF программный блок_2 выполняется, если условие **не** выполнено.

Синтаксис

Условный оператор

```
IF <условие>
    Блок программы                ; Выполнение при: <условие> == TRUE
ENDIF
```

Ветвление

```
IF <условие>
    Программный блок_1            ; Выполнение при: <условие> == TRUE
ELSE
    Программный блок_2            ; Выполнение при: <условие> == FALSE
ENDIF
```

Объяснение

IF:	Запускает условный оператор или ветвление.
ELSE:	Запускает альтернативный программный блок.
ENDIF:	Отмечает конец условного оператора или ветвления.
<условие>:	Логическое выражение, обработка которого дает TRUE или FALSE.

Пример: Подпрограмма смены инструмента

Программный код	Комментарий
PROC L6	Программа смены инструмента
N500 DEF INT TNR_AKTUELL	Переменная для активного номера T
N510 DEF INT TNR_VORWAHL	Переменная для предварительно выбранного номера T
	Определить актуальный инструмент
N520 STOPRE	
N530 IF \$P_ISTEST	В режиме тестирования программы ...

Программный код	Комментарий
N540 TNR_AKTUELL = \$P_TOOLNO	... из программного контекста считывается "актуальный" инструмент.
N550 ELSE	В ином случае ...
N560 TNR_AKTUELL = \$TC_MPP6[9998,1]	... выгружается инструмент шпинделя.
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	Чтение номера T предварительно выбранного инструмента на шпинделе.
N590 IF TNR_AKTUELL <> TNR_VORWAHL	Если предварительный инструмент это еще не актуальный инструмент, то ...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	... подвод к точке смены инструмента ...
N610 M206	... и выполнить смену инструмента.
N620 ENDIF	
N630 M17	

2.12.2 Бесконечный программный цикл (LOOP, ENDLOOP)

Бесконечный цикл используется в бесконечных программах. Концом цикла всегда является возврат в начало цикла.

Синтаксис

```

LOOP
...
ENDLOOP

```

Значение

LOOP:	Запускает бесконечный цикл.
ENDLOOP:	Обозначает конец цикла и вызывает возврат на начало цикла.

Пример

```

Программный код
...
LOOP
MSG ("нет активного резца инструмента")
M0
stopre
ENDLOOP
...

```

2.12.3 Счетный цикл (FOR ... TO ..., ENDFOR)

Счетный цикл используется в тех случаях, когда процесс работы должен быть повторен за фиксированное число проходов.

Синтаксис

```
FOR <переменная> = <начальное значение> TO <конечное значение>
...
ENDFOR
```

Значение

FOR:	Запускает счетный цикл.	
ENDFOR:	Обозначает конец цикла и вызывает возврат на начало цикла, пока конечное значение подсчета еще не достигнуто.	
<переменная>:	Счетная переменная, которая увеличивается от начального к конечному значению и приражается при каждом проходе на значение "1".	
	Тип	INT или REAL Примечание: Используется тип REAL, если, к примеру, для счетного цикла программируются R-параметры. Если счетная переменная имеет тип REAL, то ее значение округляется до целочисленного значения.
<начальное значение>:	Начальное значение подсчета Условие: Начальное значение должно быть меньше конечного значения.	
<конечное значение>:	Конечное значение подсчета	

Примеры

Пример 1: Переменная INTEGER или R-параметр как счетная переменная

Переменная INTEGER как счетная переменная:

Программный код	Комментарий
DEF INT iVARIABLE1	
R10=R12-R20*R1 R11=6	
FOR iVARIABLE1= R10 TO R11	; Счетная переменная = переменная INTEGER
R20=R21*R22+R33	
ENDFOR	
M30	

Параметр R как счетная переменная:

Программный код	Комментарий
R11=6	

Программный код	Комментарий
FOR R10=R12-R20*R1 TO R11 R20=R21*R22+R33 ENDFOR M30	; Счетная переменная = R-параметр (переменная Real)

Пример 2: Изготовление фиксированного числа деталей

Программный код	Комментарий
DEF INT STUECKZAHL FOR STUECKZAHL = 0 TO 100 G01 ... ENDFOR M30	; Определяет переменную типа INT с именем "STUECKZAHL". ; Запускает счетный цикл. Переменная "STUECKZAHL" увеличивается от начального значения "0" до конечного значения "100". ; Конец счетного цикла.

2.12.4 Программный цикл с условием в начале цикла (WHILE, ENDWHILE)

В цикле WHILE условие стоит в начале цикла. Пока условие выполнено, цикл WHILE выполняется.

Синтаксис

```
WHILE <условие>
...
ENDWHILE
```

Значение

WHILE:	Запускает программный цикл.
ENDWHILE:	Обозначает конец цикла и вызывает возврат на начало цикла.
<условие>:	Условие, которое должно быть выполнено, чтобы выполнить цикл WHILE.

Пример

Программный код	Комментарий
... WHILE \$AA_IW[BOHRACHSE] > -10 G1 G91 F250 AX[BOHRACHSE] = -1	; Вызов цикла WHILE при следующем условии: актуальное заданное значение WCS для оси сверления должно быть больше -10.

Программный код	Комментарий
ENDWHILE	
...	

2.12.5 Программный цикл с условием на конце цикла (REPEAT, UNTIL)

В цикле REPEAT условие стоит в конце цикла. Цикл REPEAT проходится один раз и повторяется до тех пор, пока выполнено условие.

Синтаксис

```
REPEAT
...
UNTIL <условие>
```

Значение

REPEAT:	Запускает программный цикл.
UNTIL:	Обозначает конец цикла и вызывает возврат на начало цикла.
<условие>:	Условие, которое должно быть выполнено, чтобы цикл WHILE более не выполнялся.

Пример

Программный код	Комментарий
...	
REPEAT	; Вызов цикла REPEAT.
...	
UNTIL ...	; Проверка, выполнено ли условие.
...	

2.12.6 Пример программы со вложенными управляющими структурами

Программный код	Комментарий
LOOP	
IF NOT \$P_SEARCH	; IF нет поиска кадра
G1 G90 X0 Z10 F1000	
WHILE \$AA_IM[X] <= 100	; WHILE (заданное значение X-ось <= 100)
G1 G91 X10 F500	; схема сверления
Z-5 F100	
Z5	

Программный код	Комментарий
ENDWHILE	
ELSE	; ELSE поиск кадра
MSG ("При поиске сверление не выполняется")	
ENDIF	; ENDIF
\$A_OUT[1] = 1	; следующая кондукторная плита
G4 F2	
ENDLOOP	
M30	

2.13 Координирующие команды (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

Любой канал ЧПУ принципиально может выполнять запущенную в нем программу независимо от других каналов своей группы режимов работы (ГРП). Но если в изготовлении детали одновременно задействовано несколько программ в нескольких каналах ГРП, программные процессы в разных каналах необходимо координировать посредством следующих координирующих команд.

Условие

Участвующие в координации программы каналы должны относиться к **одной и той же** группе режимов работы (ГРП).

MD10010 \$MC_ASSIGN_CHAN_TO_MODE_GROUP[<канал>] = <номер ГРП>

Имя канала вместо номера канала

В качестве параметра предварительно определенных процедур координации программ можно вместо номеров каналов использовать и внесенные в MD20000

\$MC_CHAN_NAME[<индекс канала>] имена каналов. Использование имен каналов в программах ЧПУ требует разблокирования:

MD10280 \$MN_PROG_FUNCTION_MASK, бит 1 = TRUE

Синтаксис

```
INIT(<ChanNr>, <Prog>, <AckMode>)
START(<ChanNr>, <ChanNr>, ...)
WAITM(<MarkNr>, <ChanNr>, <ChanNr>, ...)
WAITE(<ChanNr>, <ChanNr>, ...)
WAITMC(<MarkNr>, <ChanNr>, <ChanNr>, ...)
SETM(<MarkNr>, <MarkNr>, ...)
CLEARM(<MarkNr>, <MarkNr>, ...)
```

Объяснение

INIT():	Предварительно определенная процедура для выбора программы ЧПУ, которая должна выполняться в указанном канале
START():	Предварительно определенная процедура пуска выбранной в соответствующем канале программы
WAITM():	Предварительно определенная процедура для ожидания достижения метки ожидания в указанных каналах В собственном канале указанная метка ожидания устанавливается через WAITM. Предшествующий кадр завершается с точным остановом. После синхронизации метка ожидания удаляется. Одновременно возможна установка макс. 10 меток на канал.
WAITE():	Предварительно определенная процедура для ожидания завершения программы в одном или нескольких других каналах
WAITMC(): ¹⁾	Предварительно определенная процедура для ожидания достижения метки ожидания в указанных каналах В отличие от WAITM торможение осей до точного останова инициируется, только если другие каналы еще не достигли метки ожидания.

SETM () : ¹⁾	Предварительно определенная процедура для установки одной или нескольких меток ожидания для координации каналов Это не затрагивает обработку в собственном канале. SETM остается действительным и после сброса (Reset) канала и старта ЧПУ.		
CLEARM () : ¹⁾	Предварительно определенная процедура для удаления одной или нескольких меток ожидания координации каналов Это не затрагивает обработку в собственном канале. CLEARM () удаляет все метки ожидания в канале. CLEARM (0) удаляет только метку ожидания «0». CLEARM остается действительным и после сброса (Reset) канала и старта ЧПУ.		
<ChanNr>:	номер канала Номер собственного канала указывать не требуется.		
	Тип:	INT	
<Prog>:	Абсолютное или относительное указание пути (опция) + имя программы		
	Тип:	STRING	
	Правила указания пути см.: Литература Руководство по программированию «Расширенное программирование»; глава «Управление файлами и программами» > «Программная память» > «Адресация файлов программной памяти» (Страница 227)		
<AckMode>:	Режим квитирования (опция)		
	Тип:	CHAR	
	Зна- че- ния:	«N»	Без квитирования Обработка программы продолжается после отправки команды. Отправитель не уведомляется, если команда не может быть вы- полнена успешно.
		«S»	Синхронное квитирование Выполнение программы приостанавливается до тех пор, пока при- нимающий компонент не подтвердит команду. При положитель- ном квитировании выполняется следующая команда. При отрица- тельном квитировании выводится сообщение об ошибке.
<MarkNr>:	Номер метки ожидания Указание В многоканальной системе в распоряжении есть макс. 100 меток ожидания (метка ожидания 0 ... 99). В одноканальной системе в распоряжении есть только метка ожидания 0.		

1) Для спец. для пользователя коммуникации и/или координации каналов можно вставлять метки ожидания посредством SETM / CLEARM и без использования условной команды ожидания WAITMC. При этом метки ожидания сохраняют свои значения и после сброса (Reset) канала и старта ЧПУ.

Примеры

START посредством имен каналов из MD20000

- **Параметрирование**
MD10280 \$MN_PROG_FUNCTION_MASK, бит 1 = TRUE
\$MC_CHAN_NAME[0] = "BEARBEITUNG" ; Имя канала 1
\$MC_CHAN_NAME[1] = "ZUFUEHRUNG" ; Имя канала 2
- **Программирование**

Программный код	Комментарий
START (BEARBEITUNG)	; Пуск канала 1
START (ZUFUEHRUNG)	; Пуск канала 2

START через локальные «имена каналов» и переменную пользователя

Программный код	Комментарий
DEF INT MASCHINE = 1	; Определение переменной пользователя для канала 1
DEF INT LADER = 2	; Определение переменной пользователя для канала 2
...	
START (MASCHINE)	; Пуск канала 1
START (LADER)	; Пуск канала 2

START через локальные «имена каналов», переменную пользователя и спараметрированные имена каналов

Программный код	Комментарий
DEF INT chanNo1	; Определение переменной пользователя для канала 1
DEF INT chanNo2	; Определение переменной пользователя для канала 2
chanNo1 = CHAN_1	; Присваивание спараметрированного имени канала каналу 1
chanNo2 = CHAN_2	; Присваивание спараметрированного имени канала каналу 2
...	
START (chanNo1)	; Пуск канала 1
START (chanNo2)	; Пуск канала 2

INIT-команда с абсолютным указанием пути

Выбор программы /_N_MPF_DIR/_N_ABSPAN1_MPF в канале 2.

Программный код

```
INIT(2, "/_N_WKS_DIR/_N_WELLE1_WPD/_N_ABSPAN1_MPF")
```

INIT-команда с именем программы

Выбор программы с именем «MYPROG». СЧПУ ищет программу по маршруту поиска.

Программный код

```
INIT(2, "MYPROG")
```

Координация программ с WAITM

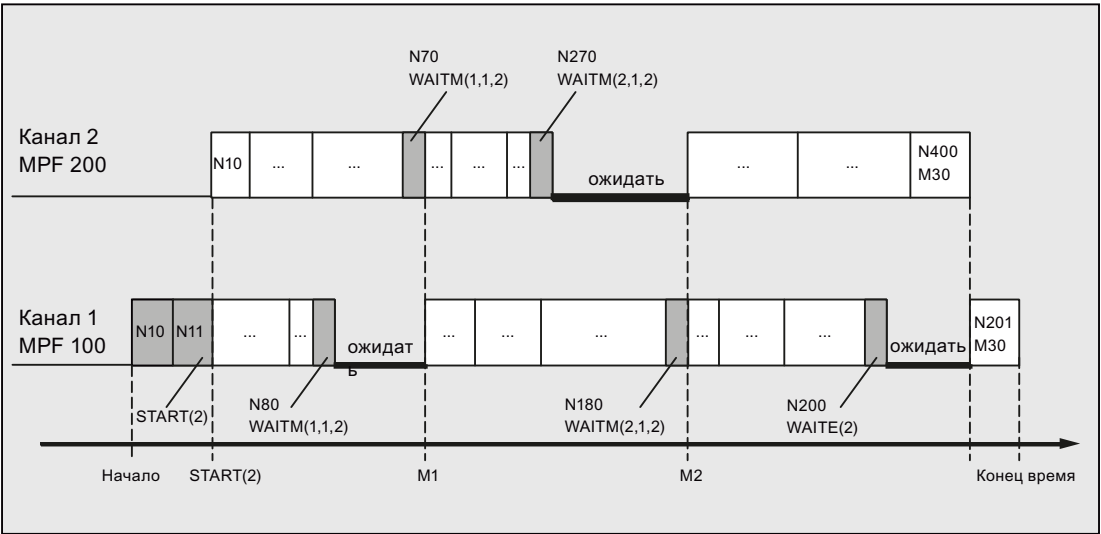
- Канал 1: Программа /_N_MPF_DIR/_N_MPF100_MPF уже выбрана и запущена.

Программный код	Комментарий
	; Программа MPF100
N10 INIT(2, "MPF200", "N")	; Выбор программы MPF200, канал 2
N11 START(2)	; Пуск канала 2
...	
N80 WAITM(1,1,2)	; Ожидание метки WAIT 1 в канале 1 и 2
N81 ...	; Канал 1, N81 и канал 2, N71 начинаются ; синхронно
...	
N180 WAITM(2,1,2)	; Ожидание метки WAIT 2 в канале 1 и 2
N181 ...	; Канал 1, N181 и канал 2, N271 начинаются ; синхронно
...	
N200 WAITE(2)	; Ожидание конца программы в канале 2
N201 ...	; N201 начинается только после конца программы ; MPF200 в канале 2
N201 M30	; Конец программы канала 1

- Канал 2: В канале 1 через кадр N10 и N20 выбирается и запускается программа MPF200_MPF для канала 2.

Программный код	Комментарий
; \$PATH=/_N_MPF_DIR	; Программа MPF200
...	
N70 WAITM(1,1,2)	Ожидание метки WAIT 1 в канале 1 и 2
N71 ...	; Канал 1, N81 и канал 2, N71 начинаются ; синхронно
...	
N270 WAITM(2,1,2)	Ожидание метки WAIT 2 в канале 1 и 2
N271 ...	; Канал 1, N181 и канал 2, N271 начинаются ; синхронно
...	
N400 M30	Конец программы канала 2

2.13 Координирующие команды (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)



Граничные условия

Асинхронное начало выполнения последующих кадров после меток WAIT

При координации каналов с помощью меток WAIT может произойти асинхронное начало выполнения последующих кадров. Это происходит, если непосредственно перед достижением общей метки WAIT в одном из синхронизируемых каналов вызывается действие, приводящее к удалению остатка пути с не явным репозиционированием (REPOSA) в этом канале.

Допущение: Текущее соотнесение осей в каналах 1 и 2

- Канал 1: оси X1 и U
- Канал 2: ось X2

Таблица 2-2 Процесс во времени в каналах 1 и 2

Канал 1	Канал 2	Описание
...	...	Любая обработка в канале 1 и 2
N100 WAITM(20,1,2)		Канал 1: достигает метки WAIT и ожидает синхронизации с каналом 2
<i>Начало обработки GETD(U):</i> • <i>Переход оси</i> • <i>Стирание остатка пути</i> • <i>REPOSA</i> <i>Конец</i>	N200 GETD(U)	Канал 2: запрашивает ось U из канала 1 Канал 1: обработка GET(U) в фоновом режиме
	N210 WAITM(20,1,2)	Канал 2: достигает метки WAIT. ⇒ На этом синхронизация каналов 1 и 2 завершена
	N220 G0 X2=100	Канал 2: начало обработки N220
N110 G0 X1=100		Канал 1: Смещенное по времени начало обработки N110

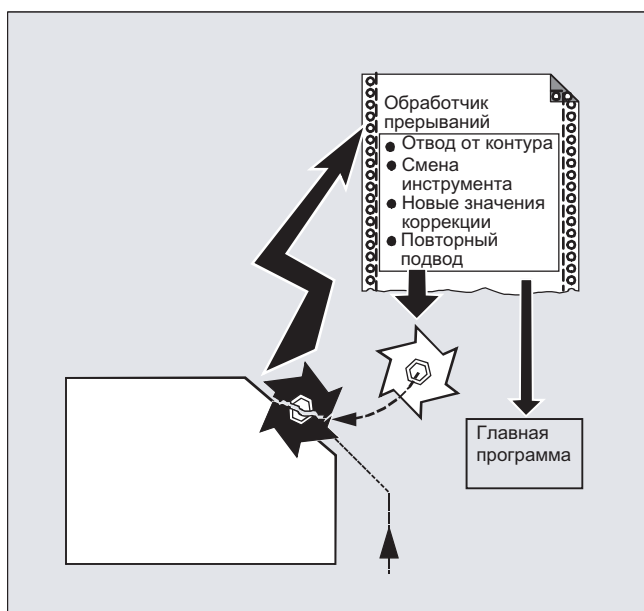
2.14 Обработчик прерываний (ASUP)

2.14.1 Функция обработчика прерываний

Примечание

Попеременно встречающиеся в описании ниже понятия "Асинхронная подпрограмма (ASUP)" и "Обработчик прерываний" обозначают одну и ту же функциональность.

Функция обработчика прерываний будет пояснена на типичном примере:



При обработке происходит поломка инструмента. Следствием этого является сигнал, останавливающий текущий процесс обработки и одновременно запускающий подпрограмму – так называемый обработчик прерываний. В этой подпрограмме находятся все операторы, которые должны быть выполнены в этом случае.

После выполнения подпрограммы (и восстановления готовности к работе), СЧПУ возвращается к главной программе и продолжает обработку – в зависимости от команды `REPOS` – в месте прерывания (см. "Повторный подвод к контуру (Страница 563)").

ОСТОРОЖНО

Опасность столкновений

Если в подпрограмме команда `REPOS` не запрограммирована, то позиционирование выполняется на конечную точку кадра, следующего за прерванным кадром.

Литература

Описание функций "Основные функции"; ГПП, канал, программный режим, параметры Reset (K1), глава: "Асинхронные подпрограммы (ASUP), обработчики прерываний"

2.14.2 Создание обработчика прерываний

Создание обработчика прерываний как подпрограммы

Обработчик прерываний при определении обозначается как подпрограмма.

Пример:

Программный код	Комментарий
PROC ABHEB_Z	; Имя программы "ABHEB_Z"
N10 ...	; Затем следуют кадры ЧПУ.
...	
N50 M17	; В заключении конец программы и возврат в главную программу.

Архивирование модальных G-команд (Save)

Обработчик прерываний при определении может быть обозначен с SAVE.

Следствием атрибута SAVE является сохранение активных модальных G-команд перед вызовом обработчика прерываний и их повторная активация после завершения обработчика прерываний (см. "Подпрограммы с механизмом SAVE (SAVE) (Страница 176) ").

Таким образом можно продолжить обработку после выполнения обработчика прерываний в месте прерывания.

Пример:

Программный код
PROC ABHEB_Z SAVE
N10 ...
...
N50 M17

Согласование дополнительных обработчиков прерываний (SETINT)

Внутри обработчика прерываний могут быть запрограммированы операторы SETINT (см. "Согласование и запуск обработчика прерываний (SETINT)" (Страница 136)), тем самым предварительно активируя дополнительные обработчики прерываний. Запуск выполняется только через вход.

Литература

Дополнительную информацию по созданию подпрограмм см. главу "Техника подпрограмм, техника макросов".

2.14.3 Согласование и запуск обработчика прерываний (SETINT, PRIO, BLSYNC)

СЧПУ имеет несколько быстрых входов (вход 1 ... 8), каждый из которых вызывает прерывание (1 ... 8). Каждому прерыванию можно с помощью команды `SETINT` сопоставить приоритет и обработчик прерываний. Если прерывание вызывается установкой быстрого входа, то текущая обработка в канале прерывается и запускается обработчик прерываний.

Приоритет прерывания

Если в какой-либо программе обработки детали прерывания назначаются нескольким входам, нужно присвоить прерываниям разные приоритеты.

Прерыванию можно присвоить одно из значений приоритета 1 ... 128. Значение приоритета 1 соответствует высшему, 128 – низшему приоритету.

Синтаксис

```
SETINT (<n>) <ИМЯ>
SETINT (<n>) PRIO=<значение> <ИМЯ>
SETINT (<n>) PRIO=<значение> <ИМЯ> BLSYNC
SETINT (<n>) PRIO=<значение> <ИМЯ> LIFTFAST
```

Значение

SETINT (<n>):	Сигналу прерывания <n> назначается программа ЧПУ (ASUP) <имя>. Назначенный обработчик прерываний запускается, как только распознается сигнал прерывания <n> == 1. Примечание: Если сигналу прерывания <n> назначается другой обработчик прерываний, предыдущее назначение перестает действовать.	
<n>:	Номер сигнала прерывания	
	Тип:	INT
	Диапазон значений:	1 ... 32
PRIO=:	Приоритет прерывания (опционально)	
<значение>:	(опция) Значение приоритета	
	Тип:	INT
	Диапазон значений:	1 ... 128 (1 ⇒ высший приоритет)
<ИМЯ>:	Имя программы ЧПУ (ASUP)	

BLSYNC:	(опция) BLSYNC действует таким образом, что после вызова прерывания сначала ожидается завершение обработки актуального кадра. Только после этого выполняется обработчик прерываний.
LIFTFAST:	(опция) LIFTFAST действует таким образом, что после вызова прерывания сначала происходит быстрый отвод (см. главу «Быстрый отвод от контура (SETINT LIFTFAST, ALF) (Страница 139)»). Только после этого выполняется обработчик прерываний.

Граничные условия

Правила обработки прерываний

1. Для каждого прерывания, которое не может обрабатываться сразу или в данный момент уже находится в обработке, сохраняется один дополнительный запрос прерывания. Остальные запросы этого прерывания утрачиваются.
2. Если в данный момент обрабатывается прерывание и вызывается еще одно прерывание с более высоким приоритетом, то оно прерывает менее приоритетное прерывание. После завершения более приоритетного прерывания продолжается обработка менее приоритетного прерывания. Если во время обработки более приоритетного прерывания поступают ещё запросы на менее приоритетное прерывание, то сохраняется один запрос. Остальные утрачиваются.
3. Если в данный момент обрабатывается прерывание и вызывается еще одно прерывание с более высоким приоритетом, то оно прерывает менее приоритетное прерывание. Более приоритетное прерывание обрабатывается. Если снова вызывается более приоритетное прерывание, то текущее прерывание прерывается, и обрабатывается более приоритетное прерывание. Максимально возможны шесть активных уровней прерывания. Один обрабатываемый в данный момент уровень прерывания и пять ожидающих уровней прерывания. Для каждого активного уровня прерывания сохраняется не более одного дополнительного запроса прерывания. Все остальные запросы прерывания утрачиваются. Запрос прерывания утрачивается и в том случае, если он направляется на остальные уровни прерывания (уровень прерывания ≥ 7).

Примеры

Пример 1: Согласование обработчиков прерываний и установка приоритета

Программный код	Комментарий
N20 SETINT(3) PRIO=1 ABHEB_Z	; IF вход 3 == 1 ; THEN запуск обработчика прерываний «ABHEB_Z»
N30 SETINT(2) PRIO=2 ABHEB_X	; IF вход 2 == 1 ; THEN запуск обработчика прерываний «ABHEB_X»

Программы последовательно выполняются в очередности значений приоритета при одновременной готовности входов: сначала "ABHEB_Z", потом "ABHEB_X".

Пример 2: Новое согласование обработчика прерываний

Программный код	Комментарий
N20 SETINT (3) PRIO=2 ABHEB_Z	; IF вход 3 == 1
...	; THEN запуск обработчика прерываний «ABHEB_Z»
N80 SETINT (3) PRIO=1 ABHEB_X	; IF вход 3 == 1
...	; THEN запуск обработчика прерываний «ABHEB_X»

2.14.4 Деактивация/повторная активация согласования обработчика прерываний (DISABLE, ENABLE)

Оператор SETINT может быть деактивирован с помощью DISABLE и снова активирован с помощью ENABLE без потери согласования "Вход → Обработчик прерываний".

Синтаксис

DISABLE (<n>)
ENABLE (<n>)

Значение

DISABLE (<n>):	Команда: Деактивация согласования обработчика прерываний входа <n>	
ENABLE (<n>):	Команда: Реактивация согласования обработчика прерываний входа <n>	
<n>:	Параметры: Номер сигнала прерывания	
	Тип:	INT
	Диапазон значений:	1 ... 32

Пример

Программный код	Комментарий
N20 SETINT (3) PRIO=1 ABHEB_Z	; при включении входа 3 должен запуститься
...	; обработчик прерываний «ABHEB_Z».
N90 DISABLE (3)	; Оператор SETINT из N20 деактивируется.
N130 ENABLE (3)	; Оператор SETINT из N20 снова активируется.
...	

2.14.5 Удаление согласования обработчика прерываний (CLRINT)

Заданное через SETINT назначение сигнала прерывания программе ЧПУ (ASUP) можно удалить с помощью CLRINT.

Синтаксис

CLRINT (<n>)

Значение

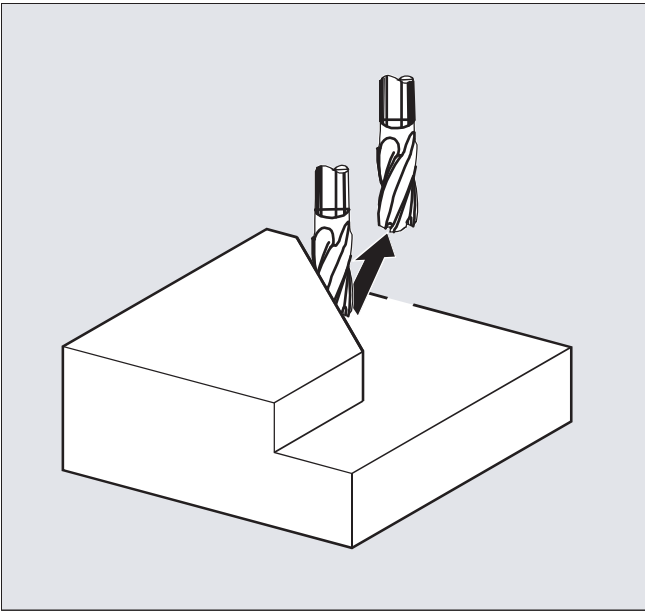
CLRINT (<n>):	Команда: Удаление назначения сигнала прерывания <n> программе ЧПУ (ASUP) <n>, определенной с помощью SETINT	
<n>:	Параметры: Номер сигнала прерывания	
	Тип:	INT
	Диапазон значений:	1 ... 32

Пример

Программный код	Комментарий
N20 SETINT(3) PRIO=2 ABHEB_Z	
...	
N50 CLRINT(3)	; Согласование между входом "3" и обработчиком прерываний "ABHEB_Z" удалено.

2.14.6 Быстрый отвод от контура (SETINT LIFTFAST, ALF)

При операторе SETINT с LIFTFAST при срабатывании входа инструмент движется от контура детали с быстрым отводом.



Дальнейший процесс зависит от того, содержит ли оператор `SETINT` наряду с `LIFTFAST` обработчик прерываний:

С обработчиком прерыва-**После** быстрого отвода выполняется обработчик прерываний.

Без обработчика прерываний- Обработка после быстрого отвода останавливается с ошибкой.

Синтаксис

```
SETINT(<n>) PRIO=1 LIFTFAST
SETINT(<n>) PRIO=1 <ИМЯ> LIFTFAST
```

Объяснение

SETINT(<n>) :	Команда: Согласовать вход <n> с обработчиком прерываний. Согласованный обработчик прерываний запускается при срабатывании входа <n>.	
<n>:	Параметры: Номер входа	
	Тип:	INT
	Диапазон значений:	1 ... 8
PRIO= :	Определение приоритета	
<значение>:	Значение приоритета	
	Диапазон значений:	1 ... 128
	Приоритет 1 соответствует высшему приоритету.	
<ИМЯ>:	Имя подпрограммы (обработчика прерываний), которая должна быть выполнена.	
LIFTFAST:	Команда: Быстрый отвод от контура	
ALF=... :	Команда: Программируемое направление перемещения (стоит в кадре движения) По возможностям программирования с ALF см. тему " Направление перемещения при быстром отводе от контура (Страница 142) ".	

Граничные условия

Поведение при активном фрейме с отражением

При определении направления отвода проверяется, активен ли фрейм с отражением. В этом случае направление отвода относительно направления касательной изменяется справа налево. Составляющие направления в направлении инструмента не отражаются. Такое поведение активируется через установку MD:

```
MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE
```

Пример

Сломанный инструмент должен быть автоматически заменен однотипным инструментом. После обработка продолжается с новым инструментом.

Главная программа:

Главная программа	Комментарий
N10 SETINT(1) PRIO=1 W_WECHS LIFTFAST	; При включении входа 1 инструмент сразу же отводится быстрым отводом (код Nr. 7 для коррекции радиуса инструмента G41) от контура. После выполняется обработчик прерываний "W_WECHS".
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	
N50 G41 G1 X16 Y16 F200	
N60 Y35	
N70 X53 Y65	
N90 X71.5 Y16	
N100 X16	
N110 G40 G0 Z100 M30	

Подпрограмма:

Подпрограмма	Комментарий
PROC W_WECHS SAVE	; Подпрограмма с сохранением актуального рабочего состояния
N10 G0 Z100 M5	; Позиция смены инструмента, стоп шпинделя
N20 T11 M6 D1 G41	; Смена инструмента
N30 REPOS L RMBBL M3	; Повторный подвод к контуру и возврат в главную программу (программируется в одном кадре)

2.14.7 Направление перемещения при быстром отводе от контура

Движение отвода

Плоскость движения отвода определяется следующими G-командами:

- **LFTXT**
Плоскость движения отвода определяется из касательной к траектории и направления инструмента (стандартная настройка).
- **LFWP**
Плоскость движения отвода — это активная рабочая плоскость, выбираемая G-командами G17, G18 или G19. Направление движения отвода не зависит от касательной к траектории. Таким образом можно запрограммировать параллельный оси быстрый отвод.
- **LFPOS**
Отвод названной с помощью `POLFMASK / POLFMLIN` оси на запрограммированную с `POLF` абсолютную позицию оси.
`ALF` не влияет на направление отвода для нескольких осей, а также для нескольких осей в линейной связи.

Литература:

Руководство по программированию "Основы"; глава: "Быстрый отвод при резьбонарезании"

Программируемое направление перемещения (`ALF=...`)

В плоскости движения отвода с помощью `ALF` программируется направление с дискретным шагом в 45 градусов.

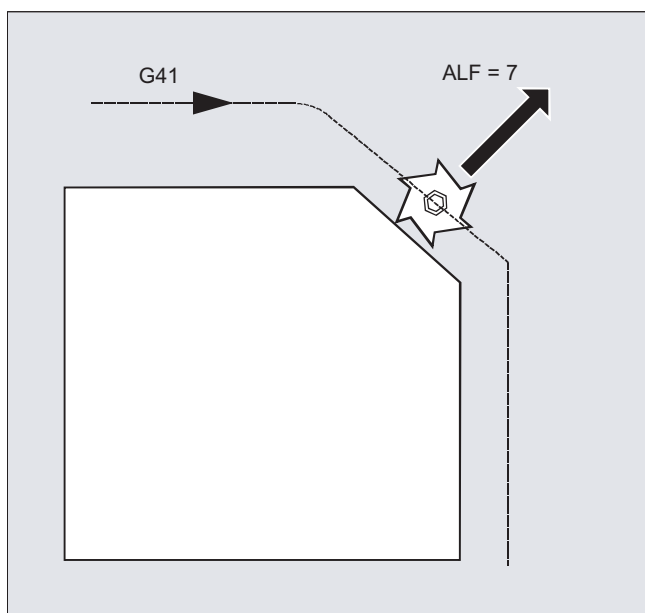
Возможные направления перемещения сохранены в СЧПУ под специальными кодовыми номерами, по которым они могут вызываться.

Пример:

Программный код

```
N10 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST
ALF=7
```

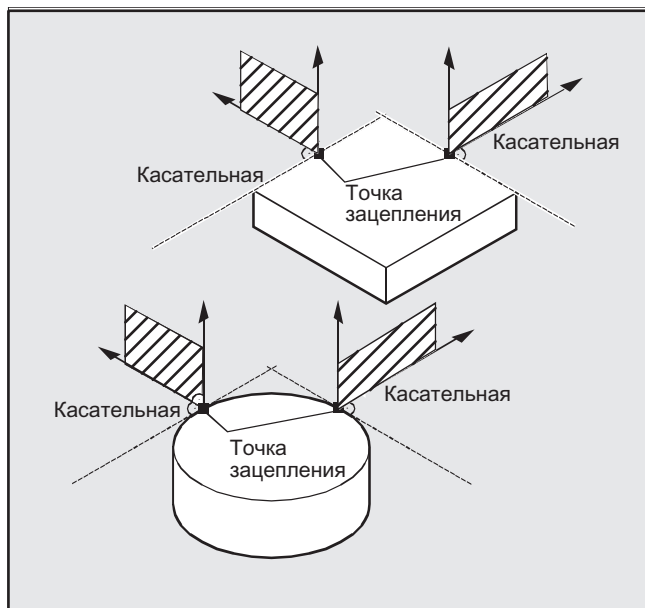
Инструмент движется с включенной G41 (направление обработки слева от контура) вертикально от контура.



Базовая плоскость для описания направлений перемещения при LFTXT

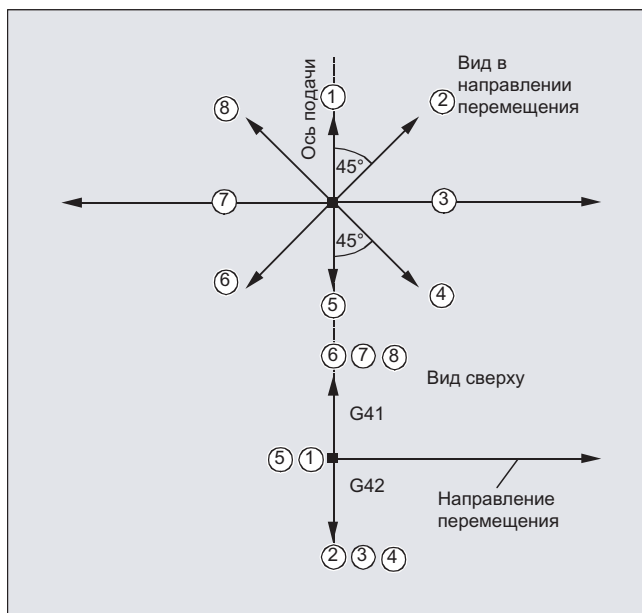
В точке касания инструмента на запрограммированном контуре создается плоскость, служащая базой для указания движения отвода с соответствующим кодовым номером.

Базовая плоскость создается из продольной оси инструмента (направление подачи) и вектора к этой оси вертикального к касательной в точке касания инструмента на контуре.



Кодовые номера с направлениям перемещения при LFTXT

Исходя из базовой плоскости, на рисунке ниже указаны кодовые номера с направлениями перемещения.



Для ALF=1 определен отвод в направлении инструмента.

При ALF=0 функция "Быстрый отвод" отключена.

⚠ ОСТОРОЖНО

Опасность столкновений

При включенной коррекции радиуса инструмента:

- при G41 кодировки 2, 3, 4
- при G42 кодировки 6, 7, 8

не должны использоваться, т.к. в этих случаях произошел бы подвод инструмента к контуру и столкновение с деталью.

Кодовые номера с направлениям перемещения при LFWP

При LFWP направление в рабочей плоскости получается согласно следующему согласованию:

- G17: плоскость X/Y
ALF=1: отвод в направлении X
ALF=3: отвод в направлении Y
- G18: плоскость Z/X
ALF=1: отвод в направлении Z
ALF=3: отвод в направлении X
- G19: плоскость Y/Z
ALF=1: отвод в направлении Y
ALF=3: отвод в направлении Z

2.14.8 Процесс движения для обработчиков прерываний

Обработчик прерываний без LIFTFAST

Движения осей затормаживаются на траектории до состояния покоя. После запускается обработчик прерываний.

Позиция покоя сохраняется как позиция прерывания и при REPOS с RMIBL в конце обработчика прерываний к ней выполняется подвод.

Обработчик прерываний с LIFTFAST

Движения осей затормаживаются на траектории. Одновременно движение LIFTFAST выполняется как наложенное движение. После остановки движения по траектории и движения LIFTFAST запускается обработчик прерываний.

В качестве позиции прерывания сохраняется позиция на контуре, на которой было запущено движение LIFTFAST и из-за этого произошел выход из траектории.

Поведение обработчика прерываний с LIFTFAST и ALF=0 идентично обработчику прерываний без LIFTFAST.

Примечание

Значение, на которое геометрические оси при быстром отводе отводятся от контура, может быть установлено через машинные данные.

2.15 Переход оси, переход шпинделя (RELEASE, GET, GETD)

Интерполяция одной или нескольких осей или шпинделей возможна только в одном канале. Если необходима попеременная работа одной оси в двух различных каналах (к примеру, устройство автоматической смены палет), то она сначала должна быть разрешена в актуальном канале и после принята в другой канал. Ось переходит между каналами.

Расширения перехода оси

Возможен переход оси/шпинделя с остановкой предварительной обработки и синхронизацией между предварительной обработкой и главным ходом или, в качестве альтернативы, и без остановки предварительной обработки. Кроме этого, переход оси возможен и через

- Вращение осевого контейнера AXCTSWE или AXCTWED посредством не явной GET/GETD.
- Фрейм с вращением, если посредством этого эта ось соединена с другими осями.
- Синхронные действия, см. Синхронные действия движения, "Переход оси RELEASE, GET".

Изготовитель станка

Следовать указаниям изготовителя станка. Через конфигурируемые машинные данные одна ось должна быть однозначно определена для перехода оси во всех каналах и изменение поведения перехода оси должно быть возможным через машинные данные.

Синтаксис

RELEASE (имя оси, имя оси, ...) или RELEASE (S1)

GET (имя оси, имя оси, ...) или GET (S2)

GETD (имя оси, имя оси, ...) или GETD (S3)

С помощью GETD (GET Directly) ось напрямую забирается из другого канала. Это означает, что для этой GETD не требуется программирования подходящей RELEASE в другом канале. Но это означает и установку иной коммуникации каналов (к примеру, метки ожидания).

Объяснение

RELEASE (имя оси, имя оси, ...):	Разрешение оси(ей)
GET (имя оси, имя оси, ...):	Получение оси(ей)
GETD (имя оси, имя оси, ...):	Прямое получение оси(ей)
Имя оси:	Согласование оси в системе: AX1, AX2, ... или указание имен осей станка
RELEASE (S1) :	Разрешение шпинделей S1, S2, ...
GET (S2) :	Получение шпинделей S1, S2, ...
GETD (S3) :	Прямое получение шпинделей S1, S2, ...

Требование GET без остановки предварительной обработки

Если после требования GET **без** остановки предварительной обработки ось снова разрешается с RELEASE (ось) или WAITP (ось), то последующая GET приводит к GETC остановкой предварительной обработки.

**ОСТОРОЖНО****Согласование осей изменено**

Полученная с GET ось или шпиндель остается согласованной с этим каналом и после клавишного или программного RESET.

При новом старте программы необходимо программно-техническое согласование переданных осей или шпинделей, если ось необходима в своем базовом канале.

При POWER ON она согласуется с зафиксированным в машинных данных каналом.

Примеры**Пример 1: Переход оси между двумя каналами**

Из 6 осей в канале 1 для обработки используются: 1-я, 2-я, 3-я и 4-я ось.

5-я и 6-я ось используются в канале 2 для смены детали.

Необходим переход оси 2 между двумя каналами и ее согласование после POWER ON с каналом 1.

Программа "MAIN" в канале 1:

Программный код	Комментарий
INIT (2, "TAUSCH2")	; Выбрать программу TAUSH2 в канале 2.
N... START (2)	; Запустить программу в канале 2.
N... GET (AX2)	; Принять ось AX2.
...	
N... RELEASE (AX2)	; Разрешить ось AX2.
N... WAITM (1, 1, 2)	; Ожидание метки WAIT в канале 1 и в канале 2 для синхронизации в обоих каналах.
...	; Продолжение процесса после перехода оси.
N... M30	

Программа "TAUSCH2" в канале 2:

Программирование	Комментарий
N... RELEASE (AX2)	
N160 WAITM(1, 1, 2)	; Ожидание метки WAIT в канале 1 и в канале 2 для синхронизации в обоих каналах.
N150 GET (AX2)	; Принять ось AX2.
...	; Продолжение процесса после перехода оси.
N... M30	

Пример 2: Переход оси без синхронизации

Если синхронизация оси не требуется, то GET не создает остановки предварительной обработки.

Программирование	Комментарий
N01 G0 X0	
N02 RELEASE (AX5)	
N03 G64 X10	
N04 X20	
N05 GET (AX5)	; Если синхронизация не нужна, то это не исполняемый кадр.
N06 G01 F5000	; Не исполняемый кадр.
N07 X20	; Не исполняемый кадр, так как позиция X как в N04.
N08 X30	; Первый исполняемый кадр после N05.
...	

Пример 3: Активация перехода оси без остановки предварительной обработки

Условие: Переход оси без остановки предварительной обработки должен быть сконфигурирован через машинные данные.

Программирование	Комментарий
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP (B)	; Ось B становится нейтральной осью.
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	; Ось не инициирует остановку предварительной обработки/REORG.
N041 G4 F2	
N050 M5	
N099 M30	

Если шпиндель или ось B непосредственно после кадра N023 перемещается как **ось PLC**, к примеру, на 180 градусов и назад на 1 градус, то эта ось снова становится нейтральной осью и не вызывает остановки предварительной обработки в кадре N40.

Дополнительная информация**Условия для перехода оси**

- Ось через машинные данные должна быть определена во всех каналах, которые хотят ее использовать.
- Через спец. для **оси** машинные данные необходимо установить, с каким каналом ось должна быть согласована после POWER ON.

Описание**Разрешить ось: RELEASE**

При разрешении оси учитывать:

1. Ось не должна участвовать в трансформации.
2. В случае соединений осей (тангенциальное управление), все оси структуры должны быть разрешены.
3. Переход конкурирующей позиционирующей оси в этом состоянии невозможен.
4. В случае мастер-оси Gantry переходят и все ведомые оси.
5. В соединениях осей (буксировка, соединение по главному значению, электронный редуктор) возможно разрешение только ведущей оси соединения.

Передача оси: GET

С помощью этой команды запускается сам процесс перехода оси. Ответственность за ось полностью лежит на канале, в котором была запрограммирована команда.

Действие GET:

Переход оси с синхронизацией:

Синхронизаций оси всегда необходима в тех случаях, когда она была промежуточно согласована в другом канале или с PLC, и перед GET не была выполнена синхронизация через "WAITP", G74 или стирание остатка пути.

- Выполняется остановка предварительной обработки (как при STOPRE).
- Обработка прерывается до тех пор, пока переход не будет выполнен полностью.

Автоматическая "GET"

Если ось принципиально имеется в канале, но не доступна в настоящий момент как "Ось канала", то автоматически выполняется GET. Если ось(и) уже синхронизированы, то остановка предварительной обработки не создается.

Возможность изменения поведения перехода оси

Момент передачи осей может быть установлен через машинные данные следующим образом:

- Автоматический переход оси выполняется между двумя каналами и тогда, когда ось через WAITP была переведена в нейтральное состояние (прежнее поведение)
- При требовании вращения осевого контейнера все согласованные с исполняемым каналом оси осевого контейнера посредством не явной GET или GETD передаются в канал. Последующий переход оси снова разрешается только после завершения вращения осевого контейнера.

2.15 Переход оси, переход шпинделя (RELEASE, GET, GETD)

- После вставленного промежуточного кадра в главном ходе проверяется, необходима ли реорганизация или нет. Только если состояния осей этого кадра **не** совпадают с актуальными состояниями осей, то необходима реорганизация.
- Вместо кадра GET с остановкой предварительной обработки и синхронизацией между предварительной обработкой и главным ходом, возможен и переход оси и без остановки предварительной обработки. В этом случае создается только один промежуточный кадр с требованием GET. В главном ходе при выполнении этого кадра проверяется, совпадают ли состояния оси в кадре с актуальными состояниями оси.

Дополнительную информацию по функциональности перехода оси или шпинделя см. Описание функций - Дополнительные функции; ГРР, каналы, переход оси (K5).

2.16 Передача оси в другой канал (АХТОСНАН)

С помощью языковой команды АХТОСНАН ось может быть запрошена для передачи ее в другой канал. Ось может быть переведена в соответствующий канал как из программы обработки детали ЧПУ, так и из синхронного действия.

Синтаксис

АХТОСНАН (имя оси,номер канала[,имя оси,номер канала[,...]])

Значение

Элемент	Описание
АХТОСНАН:	Запросить ось для определенного канала
Имя оси:	Согласование оси в системе: X, Y, O или указание участвующих имен осей станка. Исполняемый канал не должен быть собственным каналом и это также не должен быть канал, который в настоящий момент имеет право интерполяции для оси
Номер канала:	Номер канала, с которым должна быть согласована ось

Примечание

Конкурирующая позиционирующая ось и контролируемая исключительно PLC ось

Ось PLC как конкурирующая позиционирующая ось не может менять канал. Контролируемая исключительно с PLC ось не может быть согласована с программой ЧПУ.

Литература

Описание функций - Дополнительные функции; Позиционирующие оси (P2)

Пример

АХТОСНАН в программе ЧПУ

Оси X и Y известны в 1-м канале и во 2-м канале. В настоящий момент канал 1 имеет право интерполяции и в канале 1 запускается следующая программа:

Программный код	Комментарий
N110 АХТОСНАН(Y, 2)	; Переместить ось Y во 2-й канал.
N111 M0	
N120 АХТОСНАН(Y, 1)	; Снова получить ось Y (нейтральная).
N121 M0	
N130 АХТОСНАН(Y, 2, X, 2)	; Переместить ось Y и ось X во 2-й канал (нейтральные оси).
N131 M0	
N140 АХТОСНАН(Y, 2)	; Переместить ось Y во 2-й канал (программа ЧПУ).
N141 M0	

Дополнительная информация

АХТОСНАН в программе ЧПУ

При этом только при запросе оси для программы ЧПУ в собственном канале выполняется GET и тем самым происходит ожидание фактического изменения состояния. Если ось запрашивается для другого канала и если она должна стать нейтральной осью в собственном канале, только тогда запрос соответственно устанавливается.

АХТОСНАН из синхронного действия

Если ось запрашивается для собственного канала, то АХТОСНАН из синхронного действия отображается на GET из синхронного действия. В этом случае ось при первом запросе для собственного канала становится нейтральной осью. При втором запросе ось согласуется с программой ЧПУ аналогично запросу GET в программе ЧПУ. Касательно запроса GET из синхронного действия см. главу "Синхронные действия движения".

2.17 Активация машинных данных (NEWCONF)

Команда `NEWCONF` активирует все машинные данные. Функция может быть активирована и на интерфейсе управления HMI посредством нажатия программной клавиши "Активировать MD".

При выполнении функции "NEWCONF" происходит не явная остановка предварительной обработки, т. е. движение по траектории прерывается.

Синтаксис

`NEWCONF`

Объяснение

<code>NEWCONF:</code>	Команда активации всех машинных данных с условием активации "NEW_CONFIG"
-----------------------	--

Выполнение NEWCONF из программы обработки детали с выходом за границы канала

Если осевые машинные данные изменяются из программы обработки детали и после активируются с `NEWCON`, то `NEWCONF` активирует только машинные данные, вызывающие изменения для канала программы обработки детали.

Примечание

Для надежной активации всех изменений, команда `NEWCONF` должен быть выполнен в каждом канале, где сейчас вычисляются и затронутые измененными машинными данными оси или функции.

При `NEWCONF` осевые машинные данные не активируются.

Для контролируемых с PLC осей необходимо выполнить осевой `RESET`.

Пример

Фрезерная обработка: Обработка позиции сверления с помощью различных технологий

Программный код	Комментарий
<code>N10 \$MA_CONTOUR_TOL[AX]=1.0</code>	; Изменить машинные данные.
<code>N20 NEWCONF</code>	; Активировать машинные данные.
...	

2.18 Записать файл (WRITE)

С помощью команды `WRITE` кадры/данные из программы ЧПУ могут быть записаны в конец файла (журнала), находящегося в пассивной файловой системе или во внешней программной памяти. Это может быть и обрабатываемая в настоящий момент программа.

Примечание

Записываемый посредством команды `WRITE` файл создается заново, если он не существует в программной памяти.

Условие

Актуальная установленная степень защиты должна быть равна или быть выше права `WRITE` файла. Если это не так, то доступ отклоняется с сообщением об ошибке (возвращаемое значение переменной ошибки = 13).

Синтаксис

```
DEF INT <ошибка>
...
WRITE (<ошибка>,"<имя файла>" / "<ExtG>", "<кадр/данные>")
```

Значение

WRITE:	Команда добавления кадра или данных в конце указанного файла		
<ошибка>:	Параметр 1: Переменная для возврата значения ошибки		
	Тип:	INT	
	Значение:	0	нет ошибок
		1	путь не разрешен
		2	путь не найден
		3	файл не найден
		4	неправильный тип файла
		10	файл заполнен
		11	файл используется
		12	нет свободных ресурсов
		13	нет прав доступа
		14	отсутствие или неудача <code>EXTOPEN</code> для устройства вывода
		15	ошибка записи на внешнее устройство
		16	запрограммирован недействительный внешний путь

<имя файла>:	Параметр 2: Имя файла, в котором должен быть добавлен указанный кадр или указанные данные	
	Тип:	STRING
	Перед собственно именем файла можно указывать абсолютный путь. Если путь не указывается, то поиск файла осуществляется в актуальной директории (=директория выбранной программы). Правила указания пути см. "Адресация файлов программной памяти (Страница 227)".	
<ExtG>:	Если данные с помощью функции "Process DataShare" должны быть выведены на внешнее устройство/файл, то необходимо указать вместо имени файла символический идентификатор для открываемого внешнего устройства/файла.	
	Тип:	STRING
	Дополнительную информацию см. "Process DataShare - Вывод на внешнее устройство/файл (EXTOPEN, WRITE, EXTCLOSE) (Страница 700)". Примечание: Идентификатор должен быть идентичен указанному в команде EXTOPEN идентификатору.	
<кадр/данные>:	Параметр 3: Кадр или данные, которые должны быть добавлены в указанный файл.	
	Тип:	STRING

Примечание

При записи в пассивную файловую систему или во внешнюю программную память команда WRITE не явно добавляет символ "LF" (LINE FEED = новая строка) в конце строки вывода.

Для вывода на внешнее устройство/файл через функцию "Process DataShare" это поведение не действует. Если "LF" также должен быть выведен, то это должно быть указано явно в строке вывода.

→ См. пример 3: не явный/явный "LF"!

Граничные условия

- **Макс. размер файла (→ изготовитель станка!)**
Макс. возможный размер файлов журнала в пассивной файловой системе устанавливается с машинными данными:
MD11420 \$MN_LEN_PROTOCOL_FILE
Макс. размер файла действует для всех файлов, создаваемых с помощью команды WRITE в пассивной файловой системе. При превышении выводится сообщение об ошибке и кадр или данные не сохраняются. Если памяти достаточно, то может быть создан новый файл.

Примеры

Пример 1: Команда WRITE в пассивную файловую систему без абсолютного указания пути

Программный код	Комментарий
N10 DEF INT ERROR	; Определение переменных ошибок.
N20 WRITE(ERROR, "PROT", "PROTOKOLL VOM 7.2.97")	; Записать текст "PROTOKOLL VOM 7.2.97" в файл _N_PROT_MPF.
N30 IF ERROR	;Обработка ошибок.
N40 MSG ("Ошибка команды WRITE:" << ERROR)	
N50 M0	
N60 ENDIF	
...	

Пример 2: Команда WRITE в пассивную файловую систему с абсолютным указанием пути

Программный код
...
WRITE(ERROR, "/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF", "PROTOKOLL VOM 7.2.97")
...

Пример 3: Не явный/явный "LF"

а, запись в пассивную файловую систему с не явно созданным "LF"

Программный код
...
N110 DEF INT ERROR
N120 WRITE(ERROR, "/_N_MPF_DIR/_N_MYPROTFILE_MPF", "MY_STRING")
N130 WRITE(ERROR, "/_N_MPF_DIR/_N_MYPROTFILE_MPF", "MY_STRING")
N140 M30

Результат вывода:

MY_STRING

MY_STRING

б, запись во внешний файл без не явного созданного "LF"

Программный код
...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"

Программный код

```
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING")
N250 WRITE(ERROR,DEV_1,"MY_STRING")
N260 EXTCLOSE(ERROR,DEV_1)
N270 M30
```

Результат вывода:

MY_STRINGMY_STRING

с, запись во внешний файл с явно запрограммированным "LF"

Для получения результата, идентичного а, запрограммировать следующее:

Программный код

```
...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING'ноа'")
N250 WRITE(ERROR,DEV_1,"MY_STRING'ноа'")
N260 EXTCLOSE(ERROR,DEV_1)
N270 M30
```

Результат вывода:

MY_STRING

MY_STRING

2.19 Удалить файл (DELETE)

С помощью команды `DELETE` могут удаляться любые файлы, независимо от того, созданы ли они через команду `WRITE` или нет. Файлы, созданные при более высоком уровне доступа, также могут быть удалены с `DELETE`.

Синтаксис

```
DEF INT <ошибка>
DELETE (<ошибка>,"<имя файла>")
```

Значение

DELETE:	Команда удаления указанного файла		
<ошибка>:	Переменная для возврата значения ошибки		
	Тип.	INT	
	Значе- ние:	0	нет ошибок
		1	путь не разрешен
		2	путь не найден
		3	файл не найден
		4	неправильный тип файла
		11	файл используется
		12	нет свободных ресурсов
		20	иная ошибка
<имя файла>:	Имя удаляемого файла		
	Тип:	STRING	
	Перед собственно именем файла можно указывать абсолютный путь. Если путь не указывается, то поиск файла осуществляется в актуальной директории (=директория выбранной программы). Правила указания пути см. "Адресация файлов программной памяти (Страница 227)".		

Пример

Программный код	Комментарий
N10 DEF INT ERROR	; Определение переменных ошибок.
N15 STOPRE	; Остановка предварительной обработки.
N20 DELETE (ERROR, "/_N_SPF_DIR/_N_TEST1_SPF")	; Удалить файл TEST1 в директории под-программы.
N30 IF ERROR	; Обработка ошибок.
N40 MSG ("Ошибка команды DELETE:" <<ERROR)	
N50 M0	
N60 ENDIF	

2.20 Чтение строк в файле (READ)

Команда `READ` считывает в указанном файле одну или несколько строк и помещает считанную информацию в поле типа `STRING`. Каждая считанная строка в этом поле занимает один элемент поля.

Условие

Актуальная установленная степень защиты должна быть равна или быть выше права `READ` файла. Если это не так, то доступ отклоняется с сообщением об ошибке (возвращаемое значение переменной ошибки = 13).

Синтаксис

```
DEF INT <ошибка>
DEF STRING [<длина строки>] <результат>[<n>, <m>]
READ (<ошибка>, "<имя файла>", <начальная строка>, <число строк>, <результат>)
```

Объяснение

READ:	Команда по чтению строк указанного файла и по сохранению этих строк в поле переменной.			
<ошибка>:	Переменная для возврата значения ошибки (параметры, вызываемые по ссылке)			
	Тип.	INT		
	Значе- ние:	0	нет ошибок	
		1	путь не разрешен	
		2	путь не найден	
		3	файл не найден	
		4	неправильный тип файла	
		11	файл используется	
		13	прав доступа недостаточно	
		21	Строка отсутствует (параметр <начальная строка> или <число строк> больше, чем число строк в указанном файле).	
		22	Длина поля переменной результата (<результат>) слишком маленькая.	
	23	Слишком большой диапазон строк (параметр <число строк> выбран таким большим, что считается за конец файла).		
<имя файла>:	Имя считываемого файла (параметр, вызываемый по значению)			
	Тип:	STRING		
	Перед собственно именем файла можно указывать абсолютный путь. Если путь не указывается, то поиск файла осуществляется в актуальной директории (=директория выбранной программы). Правила указания пути см. "Адресация файлов программной памяти (Страница 227)".			

<начальная строка>:	Начальная строка считываемой области файла (параметр, вызываемый по значению)		
	Тип:	INT	
	Значение:	0	Считывается указанное с помощью параметра <число строк> число строк перед концом файла.
		1 ... n	Номер первой считываемой строки.
<число строк>:	Число считываемых строк (параметр, вызываемый по значению)		
	Тип:	INT	
<результат>:	Переменная результата (параметр, вызываемый по ссылке)		
	Поле переменной, в которое помещается считанный текст.		
	Тип:	STRING (макс. длина: 255)	
	Если в параметре <число строк> указано меньше строк, чем составляет размер поля [<n>, <m>] переменной результата, то оставшиеся элементы поля не изменяются.		
	Конец строки через управляющий символ "LF" (Line Feed) или "CR LF" (Carrige Return Line Feed) не помещается в переменную результата. Считанные строки обрезаются, если строка длиннее, чем определенная длина строки. Сообщение об ошибке не появляется.		

Примечание

Загрузка двоичных файлов невозможна. Выводится ошибка "неправильный тип файла" (возвращаемое значение переменной ошибки = 4). Чтение следующих типов файлов невозможно: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Пример

Программный код	Комментарий
N10 DEF INT ERROR	; Определение переменных ошибок.
N20 DEF STRING[255] RESULT[5]	; Определение переменной результата.
N30 READ(ERROR, "/_N_CST_DIR/_N_TESTFILE_MPF", 1,5,RESULT)	;Имя файла с идентификатором домена, расширением файла и указанием пути.
N40 IF ERROR <>0	;Обработка ошибок.
N50 MSG ("FEHLER"<<ERROR<<"BEI READ-BEFEHL")	
N60 M0	
N70 ENDIF	
...	

2.21 Проверка наличия файла (ISFILE)

Командой ISFILE можно проверить, существует ли файл в программной памяти.

Синтаксис

<Результат>=ISFILE ("<Имя файла>")

Объяснение

ISFILE:	Команда для проверки наличия файла		
<Имя файла>:	Имя файла, наличие которого нужно проверить.		
	Тип:	STRING	
	Перед собственно именем файла можно указывать абсолютный путь. Если путь не указывается, то поиск файла осуществляется в текущей директории (=директория выбранной программы). Правила указания пути см. "Адресация файлов программной памяти (Страница 227)".		
<Результат>:	Переменная результата для записи результата проверки		
	Тип.	BOOL	
	Значение:	TRUE	Файл имеется
		FALSE	Файл отсутствует

Примеры

Пример 1

Программный код	Комментарий
N10 DEF BOOL RESULT	; Определение переменной результата.
N20 RESULT=ISFILE ("TESTFILE")	
N30 IF (RESULT==FALSE)	
N40 MSG (">ФАЙЛ ОТСУТСТВУЕТ")	
N50 M0	
N60 ENDIF	
...	

Пример 2

Программный код	Комментарий
N10 DEF BOOL RESULT	; Определение переменной результата.
N20 RESULT=ISFILE ("TESTFILE")	
N30 IF (NOT ISFILE ("TESTFILE"))	
N40 MSG ("ФАЙЛ ОТСУТСТВУЕТ")	
N50 M0	
N60 ENDIF	
...	

2.22 Выгрузка файловой информации (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Через команды FILEDATE, FILETIME, FILESIZE, FILESTAT и FILEINFO можно выгрузить определенную файловую информацию, например, дату / время последнего обращения по записи, текущий размер файла, состояние файла или сумму этих информации.

Условие

Текущая установленная степень защиты должна быть равна или быть выше права просмотра директории верхнего уровня. Если это не так, то доступ отклоняется с сообщением об ошибке (возвращаемое значение переменной ошибки = 13).

Синтаксис

FILE . . . (<Ошибка>,"<Имя файла>",<Результат>)

Объяснение

FILEDATE:	Выводит дату последнего обращения к файлу по записи	
FILETIME:	Выводит время последнего обращения к файлу по записи	
FILESIZE:	Выводит текущий размер файла	
FILESTAT:	Выводит состояние файла касательно следующих прав : • Чтение (r: read) • Запись (w: write) • Исполнение (x: execute) • Индикация (s: show) • Удаление (d: delete) Примечание: Эти уровни защиты являются специальными свойствами пассивной файловой системы. Поэтому при доступе к внешней программной памяти FILESTAT предоставляет только права доступа по умолчанию (77777).	
FILEINFO:	Выводит для файла суммарную информацию , которая может быть выгружена через FILEDATE, FILETIME, FILESIZE и FILESTAT.	
<ошибка>:	Переменная для возврата значения ошибки (параметры, вызываемые по ссылке)	
	Тип.	VAR INT
	Значение:	0 нет ошибок
		1 путь не разрешен
		2 путь не найден
		3 файл не найден
		4 неправильный тип файла
		13 прав доступа недостаточно
	22 Длина строки переменной результата (<результат>) слишком маленькая.	

2.22 Выгрузка файловой информации (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

<Имя файла>:	Имя файла, из которого должна быть выгружена файловая информация			
	Тип:	CHAR[160]		
	Перед собственно именем файла можно указывать абсолютный путь. Если путь не указывается, то поиск файла осуществляется в текущей директории (=директория выбранной программы). Правила указания пути см. "Адресация файлов программной памяти (Страница 227)".			
<Результат>:	Переменная результата (параметр Call-By-Reference) Переменная, в которую помещается запрошенная файловая информация.			
	Тип:	VAR CHAR[8]	при	FILEDATE формат: "dd.mm.yy"
		VAR CHAR[8]	при	FILETIME формат: "hh:mm:ss"
		VAR INT	при	FILESIZE Размер файлы выводится в байтах.
		VAR CHAR[5]	при	FILESTAT формат: "rwxsd" (r: read, w: write, x: execute, s: show, d: delete)
		VAR CHAR[32]	при	FILEINFO Формат: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"

Пример

Программный код	Комментарий
N10 DEF INT ERROR	; Определение переменных ошибок.
N20 STRING[32] RESULT	; Определение переменной результата.
N30 FILEINFO(ERROR, "/_N_MPF_DIR/_N_TESTFILE_MPF", RESULT)	; Имя файла с идентификатором домена, расширением файла и указанием пути.
N40 IF ERROR <> 0	; Обработка ошибок
N50 MSG ("ОШИБКА" << ERROR << "В СЛУЧАЕ КОМАНДЫ FILEINFO")	
N60 M0	
N70 ENDIF	
...	

Пример мог бы вывести в переменной результата RESULT, например, следующий результат:

"77777 12345678 26.05.00 13:51:30"

2.23 Округления с избытком (ROUNDUP)

С помощью функции "ROUNDUP" входные значения типа REAL (дробные числа с десятичной точкой) могут быть округлены до следующего большего целого числа.

Синтаксис

ROUNDUP (<значение>)

Значение

ROUNDUP:	Команда для округления входного значения
<значение>:	Входное значение типа REAL

Примечание

Входные значения типа INTEGER (целое число) возвращаются без изменений.

Примеры

Пример 1: Различные входные значения и результаты их округления

Пример	Результат округления
ROUNDUP (3.1)	4.0
ROUNDUP (3.6)	4.0
ROUNDUP (-3.1)	-3.0
ROUNDUP (-3.6)	-3.0
ROUNDUP (3.0)	3.0
ROUNDUP (3)	3.0

Пример 2: ROUNDUP в программе обработки детали ЧПУ

Программный код

```
N10 X=ROUNDUP (3.5) Y=ROUNDUP (R2+2)
N15 R2=ROUNDUP ($AA_IM[Y])
N20 WHEN X=100 DO Y=ROUNDUP ($AA_IM[X])
...
```

2.24 Техника подпрограмм

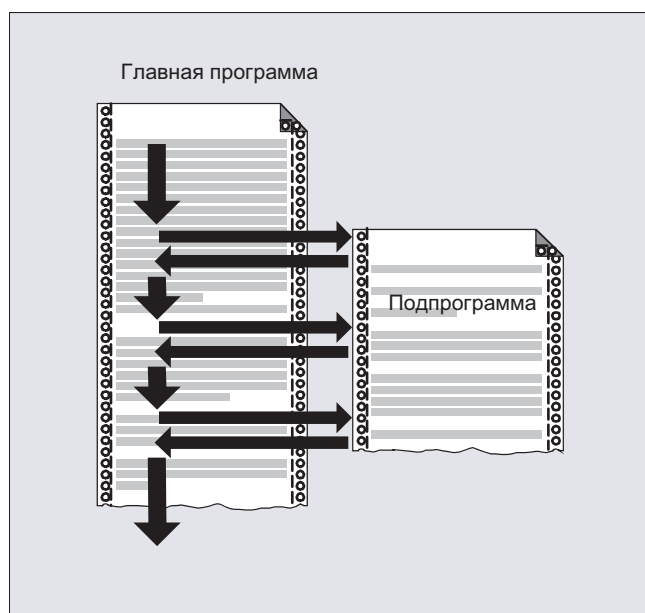
2.24.1 Общая информация

2.24.1.1 Подпрограмма

Обозначение "Подпрограмма" относится к тому времени, когда программы обработки деталей были жестко разделены на главные и подпрограммы. При этом главными программами были программы обработки деталей, которые выбирались на СЧПУ для выполнения и после запускались. Подпрограммами были программы обработки деталей, вызываемые из главной программы.

Такого жесткого разделения в современном языке ЧПУ SINUMERIK больше нет. Любая программа обработки детали в принципе может быть выбрана как главная программа и запущена или вызвана как подпрограмма из другой программы обработки детали.

Таким образом, в дальнейшем подпрограммой будет обозначаться программа обработки детали, которая вызывается из другой программы обработки детали.



Использование

Как и во всех высокоуровневых языках программирования, и в языке ЧПУ подпрограммы используются для того, чтобы поместить многократно используемые программные блоки в самостоятельные, завершенные программы.

Преимуществами подпрограмм являются:

- Увеличение наглядности и читабельности программ
- Увеличение качества через повторное использование протестированных блоков программы

- Возможность создания специфических библиотек обработки
- Экономия места в памяти

2.24.1.2 Имена подпрограмм

Правила именования

Имя подпрограммы можно выбирать произвольно, соблюдая следующие правила:

- Разрешенные символы:
 - Буквы: A ... Z, a ... z
 - Цифры: 0 ... 9
 - Знак подчеркивания: _
- Первыми двумя символами должны быть две буквы или знак подчеркивания с буквой после него.

Примечание

Если это условие выполнено, программа ЧПУ может быть вызвана из другой программы как подпрограмма путем указания имени программы. Если имя программы начинается с цифр, вызов подпрограммы возможен только через оператор CALL.

- Максимальная длина: 24 символа

Примечание

Прописные / строчные буквы

В языке ЧПУ SINUMERIK **нет** различия между прописными и строчными буквами.

Примечание

Недопустимые имена программ

Во избежание проблем с приложениями Windows **запрещено** использовать следующие имена программ:

- CON, PRN, AUX, NUL
 - COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9
 - LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9
-

Расширения внутри СЧПУ

К присвоенному при создании программы имени подпрограммы СЧПУ добавляет префикс и постфикс:

- Префикс: `_N_`
- Постфикс: `_SPF`

Использование имени программы

При использовании имени программы, например, для вызова подпрограммы, возможны любые комбинации префикса, имени программы и постфикса.

Пример:

Подпрограмма с именем программы SUB_PROG может вызываться через следующие идентификаторы:

1. SUB_PROG
2. _N_SUB_PROG
3. SUB_PROG_SPF
4. _N_SUB_PROG_SPF

Идентичность имен главных программ и подпрограмм

Если существуют главная программа (.MPF) и подпрограмма (.SPF) с одинаковым именем программы, то для однозначной идентификации при использовании имени программы в программе ЧПУ необходимо указывать и соответствующее расширение файла. В противном случае используется программа с указанным именем программы, которая будет найдена по маршруту поиска первой.

2.24.1.3 Вложенность подпрограмм

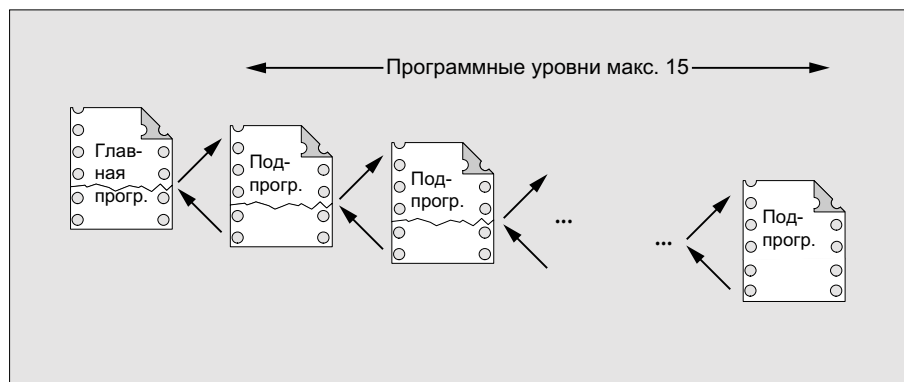
Главная программа может вызывать подпрограммы, которые в свою очередь вызывают подпрограммы. Таким образом, процессы программ вложены друг в друга. При этом каждая программа выполняется на своем программном уровне.

Глубина вложенности

Язык ЧПУ в настоящее время предлагает 16 программных уровней. Главная программа всегда выполняется на самом верхнем программном уровне 0. Подпрограмма всегда выполняется на следующем за таковым в вызове более низком программном уровне. Таким образом, программный уровень 1 это первый уровень подпрограмм.

Подразделение программных уровней:

- Программный уровень 0: уровень главной программы
- Программный уровень 1 - 15: уровень подпрограмм 1 - 15



Обработчики прерываний (ASUP)

Если в рамках обработчика прерываний вызывается подпрограмма, то она выполняется не на актуальном активном в канале программном уровне (n), а также на следующем более низком программном уровне (n+1). Для обеспечения этой возможности и на самом низком программном уровне, в комбинации с обработчиками прерываний предлагаются 2 дополнительных программных уровня (16 и 17).

Если требуется более 2 программных уровней, то это должно быть явно учтено в структурировании выполняемой в канале программы обработки детали. Т.е. в этом случае может быть задействовано макс. столько программных уровней, чтобы осталось еще достаточно программных уровней для обработки прерываний.

Если, к примеру, обработке прерываний требуется 4 программных уровня, то программа обработки детали должна быть структурирована таким образом, чтобы она занимала макс. программный уровень 13. Если после происходит прерывание, то для него доступны 4 требуемых программных уровня (14 до 17).

Циклы Siemens

Для циклов Siemens требуется 3 программных уровня. Поэтому вызов цикла Siemens должен быть выполнен самое позднее на:

- Выполнение программы обработки детали: программный уровень 12
- Обработчик прерываний: программный уровень 14

2.24.1.4 Маршрут поиска

При вызове подпрограммы без указания пути система управления выполняет поиск в имеющейся программной памяти в predetermined последовательности (см. "Маршрут поиска при вызове подпрограммы (Страница 232)").

2.24.1.5 Формальные и фактические параметры

Речь о формальных и фактических параметрах связана с определением и вызовом подпрограмм с передачей параметров.

Формальный параметр

При определении подпрограммы, передаваемые подпрограмме параметры, т.н. формальные параметры, должны быть определены с типом и именем параметра.

Тем самым формальные параметры определяют интерфейс подпрограммы.

Пример:

Программный код	Комментарий
PROC KONTUR (REAL X, REAL Y)	; Формальные параметры: X и Y оба типа REAL
N20 X1=X Y1=Y	; Перемещение оси X1 на позицию X и оси Y1 на позицию Y
...	
N100 RET	

Фактический параметр

При вызове подпрограммы ей должны быть переданы абсолютные значения или переменные, т.н. фактические параметры.

Тем самым фактические параметры при вызове заполняют интерфейс подпрограммы актуальными значениями.

Пример:

Программный код	Комментарий
N10 DEF REAL BREITE	; определение переменных
N20 BREITE=20.0	; Присваивание переменной
N30 KONTUR(5.5, BREITE)	; Вызов подпрограммы с фактическими параметрами: 5.5 и BREITE
...	
N100 M30	

2.24.1.6 Передача параметров

Определение подпрограммы с передачей параметров

Определение подпрограммы с передачей параметров осуществляется с кодовым словом PROC и полным перечнем ожидаемых от подпрограммы параметров.

Неполная передача параметров

При вызове подпрограммы не всегда должны быть явно переданы все определенные на интерфейсе подпрограммы параметры. Если параметр опускается, то для этого параметра передается стандартное значение "0".

Но для однозначного обозначения последовательности параметров, запятые в качестве разделительных символов параметров все же должны указываться всегда. Исключением является последний параметр. Если он опускается при вызове, может быть опущена и последняя запятая.

Пример:

Подпрограмма:

Программный код	Комментарий
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; Формальные параметры: X, Y и Z
...	
N100 RET	

Главная программа:

Программный код	Комментарий
PROC MAIN_PROG	
...	
N30 SUB_PROG(1.0, 2.0, 3.0)	; Вызов подпрограммы с полной передачей параметров: X=1.0, Y=2.0, Z=3.0
...	
N100 M30	

Примеры вызова подпрограммы в N30 с неполной передачей параметров:

N30 SUB_PROG(, 2.0, 3.0)	; X=0.0, Y=2.0, Z=3.0
N30 SUB_PROG(1.0, , 3.0)	; X=1.0, Y=0.0, Z=3.0
N30 SUB_PROG(1.0, 2.0)	; X=1.0, Y=2.0, Z=0.0
N30 SUB_PROG(, , 3.0)	; X=0.0, Y=0.0, Z=3.0
N30 SUB_PROG(, ,)	; X=0.0, Y=0.0, Z=0.0

ВНИМАНИЕ

Передача параметров, вызываемых по ссылке

Параметры, передаваемые по ссылке, не могут быть опущены при вызове подпрограммы.

ВНИМАНИЕ

Тип данных AXIS

Параметры типа данных AXIS не могут быть опущены при вызове подпрограммы.

Проверка передаваемых параметров

Через системную переменную \$P_SUBPAR [n] где n = 1, 2, ... в подпрограмме можно проверить, был ли параметр явно передан или опущен. Индекс n относится к последовательности формальных параметров. Индекс n = 1 относится к 1-му формальному параметру, индекс n = 2 ко 2-му формальному параметру и т.д.

Следующий сегмент программы показывает на примере для 1-го формального параметра, как может быть реализована проверка:

Программирование	Комментарий
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; Формальные параметры: X, Y и Z
N20 IF \$P_SUBPAR[1]==TRUE	; Проверка 1-го формального параметра X.
...	; Эти операции выполняются, если формальный параметр X был передан явно.
N40 ELSE	
...	; Эти операции выполняются, если формальный параметр X не был передан.
N60 ENDIF	
...	; Общие операции
N100 RET	

2.24.2 Определение подпрограммы

2.24.2.1 Подпрограмма без передачи параметров

При определении подпрограмм без передачи параметров строка определения в начале программы не нужна.

Синтаксис

```
[PROC <имя программы>]
```

```
...
```

Значение

PROC:	Оператор определения в начале программы
<имя программы>:	Имя программы

Пример

Пример 1: Подпрограмма с оператором PROC

Программный код	Комментарий
PROC SUB_PROG	; Строка определения
N10 G01 G90 G64 F1000	

Программный код	Комментарий
N20 X10 Y20	
...	
N100 RET	; возврат из подпрограммы

Пример 2: Подпрограмма без оператора PROC

Программный код	Комментарий
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; возврат из подпрограммы

См. также

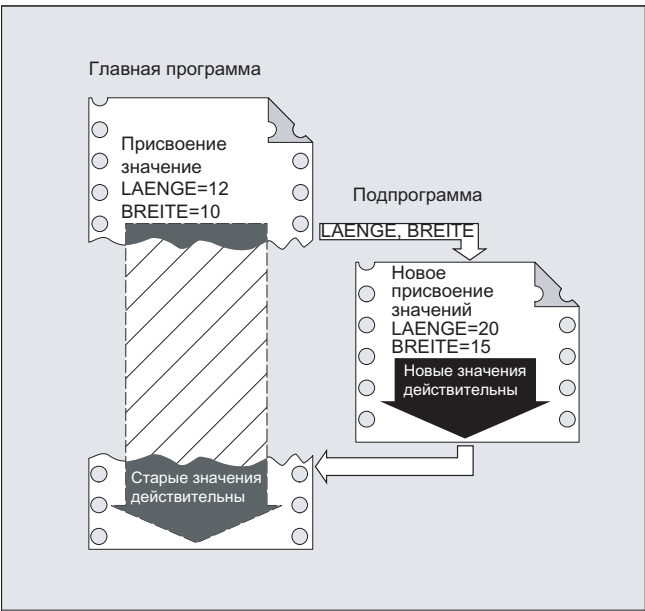
Вызовы подпрограмм без передачи параметров (Страница 199)

2.24.2.2 Подпрограмма с передачей параметров по значению (PROC)

Определение подпрограммы с передачей параметров, вызываемых по значению, выполняется кодовым словом PROC с последующим именем программы и полным перечнем параметров с типом и именем. Оператор определения должен стоять в первой строке программы.

Параметры, вызываемые по значению

Вызывающая программа при передаче параметра, вызываемого по значению, передает в подпрограмму только значение переменной. Так подпрограмма не получает прямого доступа к переменной. В результате при изменении значения параметра изменяется только значение, видимое в подпрограмме. Значение переменной, определенной в вызывающей программе, остается неизменным. Так передача параметров, вызываемых по значению, не оказывает обратного воздействия на вызывающую программу.



Синтаксис

PROC <имя программы> (<тип параметра> <имя параметра>=<Init_Wert>, ...)

Примечание

Всего может быть передано до 127 параметров.

Объяснение

PROC:	Оператор определения в начале программы
<имя программы>:	Имя программы
<тип параметра>:	Тип данных параметра (например, REAL, INT, BOOL)
<имя параметра>:	Имя параметра
<иниц_значение>:	Опциональное значение для инициализации параметра (опция) Если при вызове подпрограммы параметр не указывается, то параметру присваивается инициализирующее значение.

Примеры

Пример 1

Определение подпрограммы SUB_PROG с тремя параметрами типа REAL со значениями по умолчанию:

Программный код

```
PROC SUB_PROG (REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HIGHT=30.0)
```

Пример 2**Различные варианты вызова****Программный код**

```
PROC MAIN_PROG
    REAL PAR_1 = 100
    REAL PAR_2 = 200
    REAL PAR_3 = 300
; варианты вызова
    SUB_PROG
    SUB_PROG(PAR_1, PAR_2, PAR_3)
    SUB_PROG(PAR_1)
    SUB_PROG(PAR_1, , PAR_3)
    SUB_PROG( , , PAR_3)
N100 RET
```

См. также

Вызов подпрограммы с передачей параметров (EXTERN) (Страница 201)

2.24.2.3 Подпрограмма с передачей параметров по ссылке (PROC, VAR)

Определение подпрограммы с передачей параметров по ссылке осуществляется с кодовым словом `PROC`, с последующим именем программы и полным перечнем параметров с кодовым словом `VAR`, с типом и именем. Оператор определения должен стоять в первой строке программы. В качестве параметра можно также передавать ссылки на поля.

Параметры, вызываемые по ссылке

Вызывающая программа при передаче параметра по ссылке передает в подпрограмму не значение переменной, а ссылку (указатель) на переменную. Таким образом, подпрограмма получает прямой доступ к переменной. В результате при изменении значения параметра изменяется не только значение, видимое в подпрограмме, но и значение переменной, определенной в вызывающей программе. Таким образом, передача параметров по ссылке оказывает обратное воздействие на вызывающую программу, в том числе, после завершения подпрограммы.



Примечание

Передача параметров по ссылке необходима только тогда, когда переданная переменная была локально определена в вызывающей программе (LUD). Глобальные переменные канала или ЧПУ не должны передаваться, т.к. к ним возможен и прямой доступ из подпрограммы.

Синтаксис

PROC <имя программы> (VAR <тип параметра> <имя параметра>, ...)
PROC <имя программы> (VAR <тип поля> <имя поля> [<m>, <n>, <o>], ...)

Примечание

Всего может быть передано до 127 параметров.

Объяснение

PROC:	Оператор определения в начале программы
VAR:	Кодовое слово для передачи параметров по ссылке
<имя программы>:	Имя программы
<тип параметра>:	Тип данных параметра (например, REAL, INT, BOOL)
<имя параметра>:	Имя параметра
<тип поля>:	Тип данных элементов поля (например, REAL, INT, BOOL)
<имя поля>:	Имя поля

[<m>, <n>, <o>]:	Размер поля	
	Сейчас возможны макс. 3-мерные поля:	
	<m>:	Размер поля для 1-й размерности
	<n>:	Размер поля для 2-й размерности
	<o>:	Размер поля для 3-й размерности

Примечание

- Указанное после кодового слова `PROC` имя параметра должно совпадать с присвоенным на интерфейсе именем программы.
- С помощью полей неопределенной длины как формальных параметров подпрограммы могут обрабатывать поля переменной длины. Для этого при определении, например, двухмерного поля как формального параметра длина 1-й размерности не указывается. Но запятая должна быть записана.
Пример: `PROC <имя программы> (VAR REAL FELD[, 5])`

Пример

Определение подпрограммы с двумя параметрами как ссылка на тип `REAL`:

Программный код

```
; Параметр 1: Ссылка на тип: REAL, имя: LAENGE
; Параметр 2: Ссылка на тип: REAL, имя: BREITE
PROC SUB_PROG (VAR REAL LAENGE, VAR REAL BREITE)
```

См. также

Вызов подпрограммы с передачей параметров (`EXTERN`) (Страница 201)

2.24.2.4 Сохранить модальные функции G (SAVE)

Следствием атрибута `SAVE` является то, что активные перед вызовом подпрограммы модальные G-команды сохраняются и после завершения подпрограммы снова активируются.

ВНИМАНИЕ**Прерывание режима управления траекторией**

Если при активном режиме управления траекторией вызывается подпрограмма с атрибутом `SAVE`, то режим управления траекторией прерывается на конце подпрограммы (возврат).

Синтаксис

```
PROC <имя подпрограммы> SAVE
```


Объяснение

SAVE:	Сохранение модальных G-команд перед вызовом подпрограммы и восстановление после завершения подпрограммы
-------	---

Пример

В подпрограмме KONTUR действует модальная команда G91 (составной размер). В главной программе действует модальная команда G90 (абсолютный размер). Посредством определения подпрограммы с SAVE после завершения подпрограммы в главной программе снова действует G90.

Определение подпрограммы:

Программный код	Комментарий
PROC KONTUR (REAL WERT1) SAVE	; Определение подпрограммы с параметром SAVE
N10 G91 ...	; Модальная G-команда G91: Составной размер
N100 M17	; Конец подпрограммы

Главная программа:

Программный код	Комментарий
N10 G0 X... Y... G90	; Модальная G-команда G90: Абсолютный размер
N20 ...	
...	
N50 KONTUR (12.4)	; Вызов подпрограммы
N60 X... Y...	; Модальная G-команда G90 реактивирована через SAVE

Граничные условия

Фреймы

Поведение фреймов касательно подпрограмм с атрибутом SAVE зависит от типа фрейма и может быть установлено через машинные данные.

Литература

Описание функций "Основные функции"; Оси, системы координат, фреймы (K2), глава: "Возврат из подпрограммы с SAVE"

2.24.2.5 Подавление покадровой обработки (SBLOF, SBLON)

Подавление отдельного кадра для всей программы

Обозначенные с SBLOF программы при активной покадровой обработке выполняются целиком как один кадр, т. е. для всей программы покадровая обработка подавляется.

SBLOF стоит в строке PROC и действует до конца или отмены подпрограммы. С помощью команды возврата принимается решение, произойдет ли остановка на конце подпрограммы или нет:

Возврат с M17: Стоп на конце подпрограммы

Возврат с RET: Нет останова на конце подпрограммы

Подавление отдельного кадра внутри программы

SBLOF должна стоять одна в кадре. От этого кадра отдельный кадр отключается до:

- следующей SBLON или
- до конца активного уровня подпрограммы

Синтаксис

Подавление отдельного кадра для всей программы:

```
PROC ... SBLOF
```

Подавление отдельного кадра внутри программы:

```
SBLOF
```

```
...
```

```
SBLON
```

Значение

PROC:	Первый оператор программы
SBLOF:	Команда для отключения покадровой обработки SBLOF может стоять в кадре PROC или одна в кадре.
SBLON:	Команда для включения покадровой обработки SBLON должна стоять в собственном кадре.

Граничные условия

- Подавление отдельного кадра и индикация кадра**
 Актуальная индикация кадра может быть подавлена в циклах/подпрограммах с `DISPLOF`. Если `DISPLOF` программируется вместе с `SBLOF`, то при остановке покадровой обработки внутри цикла/подпрограммы как и прежде индицируется вызов цикла/подпрограммы.
- Подавление отдельного кадра в системной ASUP или ASUP пользователя**
 Если остановка покадровой обработки подавляется в системной ASUP или ASUP пользователя через установки в машинных данных `MD10702 $MN_IGNORE_SINGLEBLOCK_MASK` (Бит0 = 1 или Бит1 = 1), то останов покадровой обработки через программирование `SBLON` в ASUP снова может быть активирован.
 Если остановка покадровой обработки подавляется в ASUP пользователя через установки в машинных данных `MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP`, то останов покадровой обработки через программирование `SBLON` в ASUP **не** может быть снова активирован.
- Особенности подавления отдельного кадра для различных типов покадровой обработки**
 При активной покадровой обработке `SBL2` (стоп после каждого кадра программы обработки детали) в кадре `SBLON` **не** происходит останова, если в `MD10702 $MN_IGNORE_SINGLEBLOCK_MASK` (недопущение останова покадровой обработки) Бит 12 установлен на "1".
 При активной покадровой обработке `SBL3` (стоп после каждого кадра программы обработки детали и в цикле) команда `SBLOF` подавляется.

Примеры

Пример 1: Подавление отдельного кадра внутри программы

Программный код	Комментарий
N10 G1 X100 F1000	
N20 SBLOF	; Отключить отдельный кадр
N30 Y20	
N40 M100	
N50 R10=90	
N60 SBLON	; Снова включить отдельный кадр
N70 M110	
N80 ...	

Область между N20 и N60 в режиме покадровой обработки обрабатывается как один шаг.

Пример 2: Цикл должен действовать для пользователя как одна команда

Главная программа:

Программный код
N10 G1 X10 G90 F200

Программный код

```

N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
N50 M30

```

Цикл CYCLE1:**Программный код****Комментарий**

Программный код	Комментарий
N100 PROC CYCLE1 DISPLOF SBLOF	; Подавление отдельного кадра
N110 R10=3*SIN(R20)+5	
N120 IF (R11 <= 0)	
N130 SETAL(61000)	
N140 ENDIF	
N150 G1 G91 Z=R10 F=R11	
N160 M17	

Цикл CYCLE1 выполняется при активной покадровой обработке, т. е. для обработки CYCLE1 необходимо один раз нажать клавишу Start.

Пример 3: Запущенная с PLC ASUP для активации измененного смещения нулевой точки и коррекций инструмента должна быть скрыта.

Программный код

```

N100 PROC NV SBLOF DISPLOF
N110 CASE $P_UIFRNUM OF          0 GOTOF _G500
                                1 GOTOF _G54
                                2 GOTOF _G55
                                3 GOTOF _G56
                                4 GOTOF _G57
                                DEFAULT GOTOF END
N120 _G54: G54 D=$P_TOOL T=$P_TOOLNO
N130 RET
N140 _G54: G55 D=$P_TOOL T=$P_TOOLNO
N150 RET
N160 _G56: G56 D=$P_TOOL T=$P_TOOLNO
N170 RET
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO
N190 RET
N200 END: D=$P_TOOL T=$P_TOOLNO
N210 RET

```

Пример 4: При MD10702 Бит 12 = 1 останов не выполняется

Исходная ситуация:

- Покадровая обработка активна.
- MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK Бит12 = 1

Главная программа:

Программный код	Комментарий
N10 G0 X0	; Останов в этой строке программы обработки детали.
N20 X10	; Останов в этой строке программы обработки детали.
N30 CYCLE	; Созданный циклом кадр перемещения.
N50 G90 X20	; Останов в этой строке программы обработки детали.
M30	

Цикл CYCLE:

Программный код	Комментарий
PROC CYCLE SBLOF	; Подавление останова покадровой обработки
N100 R0 = 1	
N110 SBLON	; Из-за MD10702 Бит12=1 останов в этой строке программы обработки детали не выполняется.
N120 X1	; Останов в этой строке программы обработки детали.
N140 SBLOF	
N150 R0 = 2	
RET	

Пример 5: Подавление отдельного кадра при вложенности программы

Исходная ситуация:

Покадровая обработка активна.

Вложение программы:

Программный код	Комментарий
N10 X0 F1000	; Останов в этом кадре.
N20 UP1(0)	
PROC UP1(INT _NR) SBLOF	; Подавление останова покадровой обработки.
N100 X10	
N110 UP2(0)	
PROC UP2(INT _NR)	
N200 X20	
N210 SBLON	; Включение останова покадровой обработки.
N220 X22	; Останов в этом кадре.

Программный код	Комментарий
N230 UP3 (0)	
PROC UP3 (INT _NR)	
N300 SBLOF	; Подавление останова покадровой обработки.
N305 X30	
N310 SBLON	; Включение останова покадровой обработки.
N320 X32	; Останов в этом кадре.
N330 SBLOF	; Подавление останова покадровой обработки.
N340 X34	
N350 M17	; SBLOF активна.
N240 X24	; Останов в этом кадре. SBLON активна.
N250 M17	; Останов в этом кадре. SBLON активна.
N120 X12	
N130 M17	; Останов в этом кадре возврата. SBLOF оператора PROC активна.
N30 X0	; Останов в этом кадре.
N40 M30	; Останов в этом кадре.

Дополнительная информация

Блокировка отдельного кадра для асинхронных подпрограмм

Для выполнения ASUP в отдельном кадре за один шаг, необходимо запрограммировать в ASUP оператор PROC с SBLOF. Это же относится и к функции "Редактируемая системная ASUP" (MD 11610 \$MN_ASUP_EDITABLE).

Пример для "Редактируемой системной ASUP":

Программный код	Комментарий
N10 PROC ASUP1 SBLOF DISPLOF	
N20 IF \$AC_ASUP=='H200'	
N30 RET	; Без REPOS при смене режима работы.
N40 ELSE	
N50 REPOSA	; REPOS во всех прочих случаях.
N60 ENDIF	

Управление программой в отдельном кадре

В покадровой обработке пользователь может покадрово выполнять программу обработки детали. Существуют следующие типы установки:

- SBL1: отдельный кадр IPO с остановом после каждого кадра с функцией станка.
- SBL2: отдельный кадр с остановом после каждого кадра.
- SBL3: останов в цикле (через выбор SBL3 команда SBLOF подавляется).

Подавление отдельного кадра при вложенности программы

Если в подпрограмме была запрограммирована SBLOF в операторе PROC, то происходит останов на возврате из подпрограммы с M17. Тем самым не допускается исполнение

уже следующего кадра в вызывающей программе. Если в подпрограмме с SBLOF, без SBLOF в операторе PROC, активируется подавление покадровой обработки, то останов происходит только после следующего кадра функций станка вызывающей программы. Если это не желательно, то в подпрограмме еще до возврата (M17) снова необходимо запрограммировать SBLON. При возврате с RET в программу верхнего уровня останов не происходит.

2.24.2.6 Подавление актуальной индикации кадра (DISPLOF, DISPLON, ACTBLOCNO)

На индикации кадра стандартно индицируется актуальный программный кадр. В циклах или подпрограммах индикация актуального кадра может быть подавлена с помощью команды DISPLOF. Вместо актуального кадра тогда индицируется вызов цикла или подпрограммы. С помощью команды DISPLON подавление индикации кадров может быть снова отменено.

DISPLOF или DISPLON программируется в строке программы с оператором PROC и действует для всей подпрограммы и не явно для всех вызванных из этой подпрограммы подпрограмм, не содержащих команды DISPLON или DISPLOF. Это же поведение действует и для ASUP.

Синтаксис

```
PROC ... DISPLOF
PROC ... DISPLOF ACTBLOCNO
PROC ... DISPLON
```

Значение

DISPLOF:	Команда для подавления актуальной индикации кадра.	
	Размещение:	На конце строки программы с оператором PROC
	Активность:	До возврата из подпрограммы или завершения программы.
	Примечание: Если из подпрограммы с помощью команды DISPLOF вызываются другие подпрограммы, то и в этих подпрограммах актуальная индикация кадра подавляется, если в них явно не запрограммирована DISPLON.	
DISPLON:	Команда отмены подавления актуальной индикации кадров	
	Размещение:	На конце строки программы с оператором PROC
	Активность:	До возврата из подпрограммы или завершения программы.
	Примечание: Если из подпрограммы с помощью команды DISPLON вызываются другие подпрограммы, то и в этих подпрограммах отображается актуальный кадр программы, если в них явно не запрограммирована DISPLOF.	
ACTBLOCNO:	DISPLOF вместе с атрибутом ACTBLOCNO приводит к тому, что в случае ошибки выводится номер актуального кадра, в котором возникла ошибка. Это действует и в том случае, когда на более низком программном уровне запрограммирована только DISPLOF. При DISPLOF без ACTBLOCNO, напротив, индицируется номер кадра вызова цикла или подпрограммы из последнего не обозначенного с DISPLOF программного уровня.	

Примеры

Пример 1: Подавление актуальной индикации кадра в цикле

Программный код	Комментарий
PROC CYCLE (AXIS TOMOV, REAL POSITION) SAVE DISPLOF	; Подавление актуальной индикации кадра. Вместо этого должен быть показан вызов цикла, к примеру : CYCLE (X, 100.0)
DEF REAL DIFF G01	; Содержание циклов
RET	; Возврат из подпрограммы. На индикации кадра появится следующий за вызовом цикла кадр.

Пример 2: Индикация кадра при выводе ошибки

Подпрограмма SUBPROG1 (с ACTBLOCNO):

Программный код	Комментарий
PROC SUBPROG1 DISPLOF ACTBLOCNO N8000 R10 = R33 + R44 ... N9040 R10 = 66 X100 ... N10000 M17	; Запустить ошибку 12080

Подпрограмма SUBPROG2 (без ACTBLOCNO):

Программный код	Комментарий
PROC SUBPROG2 DISPLOF N5000 R10 = R33 + R44 ... N6040 R10 = 66 X100 ... N7000 M17	; Запустить ошибку 12080

Главная программа:

Программный код	Комментарий
N1000 G0 X0 Y0 Z0 N1010 N2050 SUBPROG1 N2060 ...	; Вывод ошибки = "12080 канал K1 кадр N9040 синтаксическая ошибка в тексте R10="

Программный код	Комментарий
N2350 SUBPROG2	; Вывод ошибки = "12080 канал K1 кадр N2350 синтаксическая ошибка в тексте R10="
...	
N3000 M30	

Пример 3: Отменить подавление актуальной индикации кадра

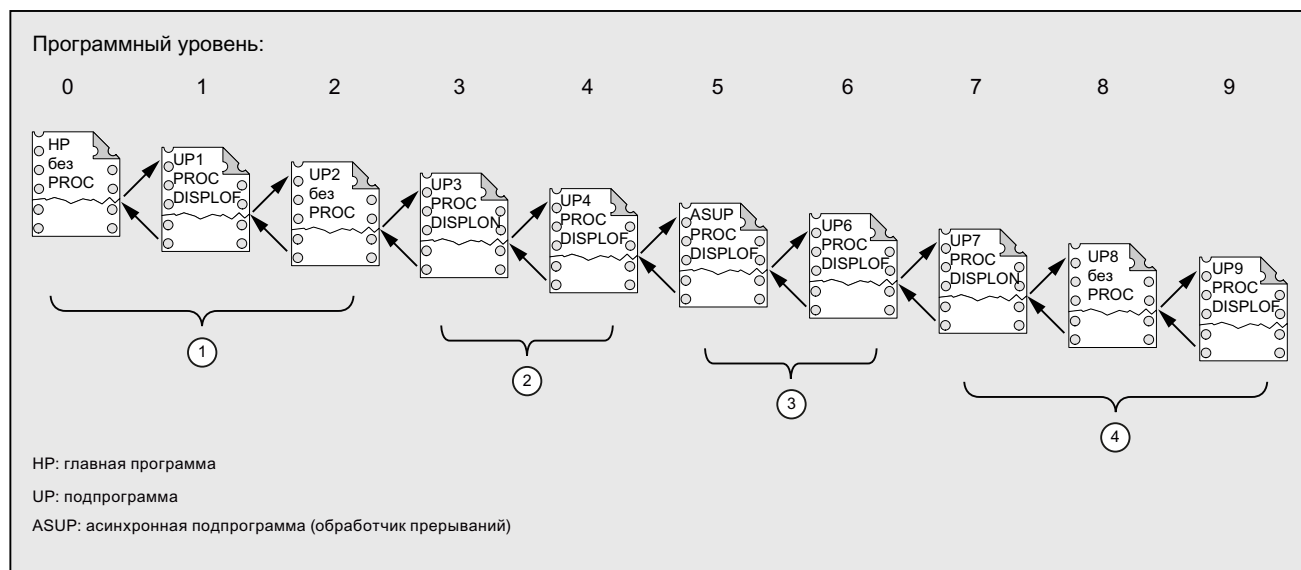
Подпрограмма SUB1 с подавлением:

Программный код	Комментарий
PROC SUB1 DISPLOF	; Подавление актуальной индикации кадров в подпрограмме SUB1. Вместо этого должен быть показан кадр с вызовом SUB1.
...	
N300 SUB2	; Вызвать подпрограмму SUB2.
...	
N500 M17	

Подпрограмма SUB2 без подавления:

Программный код	Комментарий
PROC SUB2 DISPLON	; Отменить подавление актуальной индикации кадров в подпрограмме SUB2.
...	
N200 M17	; Возврат в подпрограмму SUB1. В SUB1 актуальная индикация кадров снова подавляется.

Пример 4: Параметры индикации при различных комбинациях DISPLON/DISPLOF



- ① На актуальной индикации кадров отображаются строки программы обработки детали из программного уровня 0.
- ② На актуальной индикации кадров отображаются строки программы обработки детали из программного уровня 3.
- ③ На актуальной индикации кадров отображаются строки программы обработки детали из программного уровня 3.
- ④ На актуальной индикации кадров отображаются строки программы обработки детали из программного уровня 7/8.

2.24.2.7 Обозначить подпрограммы с подготовкой (PREPRO)

С помощью кодового слова `PREPRO` при запуске в конце строки оператора `PROC` можно обозначить все файлы.

Примечание

Этот тип подготовки программы зависит от соответственно установленных машинных данных. Просьба следовать указаниям изготовителя станка.

Литература:

Описание функций "Специальные функции"; Предварительная обработка (V2)

Синтаксис

```
PROC ... PREPRO
```

Значение

PREPRO:	Кодовое слово для обозначения всех подготовленных при запуске файлов, сохранных в директориях циклов программ ЧПУ
---------	---

Загрузка подпрограмм с подготовкой и вызов подпрограмм

Как подготовленные при запуске подпрограммы с параметрами, так и при вызове подпрограммы, обработка директорий циклов выполняется в той же последовательности:

1. _N_CUS_DIR циклы пользователя
2. _N_CMA_DIR циклы изготовителя
3. _N_CST_DIR стандартные циклы

В случае одноименных программ ЧПУ с различным оформлением активируется последний найденный оператор PROC, а другой оператор PROC игнорируется без сообщения об ошибке.

2.24.2.8 Возврат из подпрограммы M17

В конце подпрограммы стоит команда возврата M17 (или команда программы обработки детали M30). Она вызывает возврат в вызывающую программу на кадр программы обработки детали после вызова подпрограммы.

Примечание

M17 и M30 обрабатываются в языке ЧПУ равнозначно.

Синтаксис

PROC <имя программы>

...

M17/M30

Граничные условия**Влияние возврата из подпрограммы на режим управления траекторией**

Если M17 (или M30) стоит одна в кадре программы обработки детали, то из-за этого активный в канале режим управления траекторией прерывается.

Во избежание прерывания режима управления траекторией, записать M17 (или M30) и в последний кадр перемещения. Дополнительно следующие машинные данные должны быть установлены на "0":

MD20800 \$MC_SPF_END_TO_VDI = 0 (нет вывода M30/M17 на интерфейс ЧПУ/PLC)

Пример

1. подпрограмма с M17 в отдельном кадре

Программный код	Комментарий
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10	
N30 M17	; Возврат с прерыванием режима управления траекторией.

2. подпрограмма с M17 в последнем кадре перемещения

Программный код	Комментарий
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10 M17	; Возврат без прерывания режима управления траекторией.

2.24.2.9 Возврат из подпрограммы RET

В качестве эквивалента для команды возврата с M17, в подпрограмме можно использовать и команду RET. RET должна быть запрограммирована в отдельном кадре программы обработки детали. Как и M17 RET вызывает возврат в вызывающую программу на кадр программы обработки детали после вызова подпрограммы.

Примечание

Через программирование параметров, параметры возврата RET могут быть изменены (см. "Параметрируемый возврат из подпрограммы (RET ...) (Страница 189)").

Использование

Использовать оператор RET тогда, когда режим управления траекторией G64 (G641 ... G645) не должен прерываться возвратом.

Условие

Команда RET может использоваться только в подпрограммах, которые не были определены с атрибутом SAVE.

Синтаксис

```
PROC <имя программы>
...
RET
```

Пример

Главная программа:

Программный код	Комментарий
PROC MAIN_PROGRAM	; Начало программы
...	
N50 SUB_PROG	; Вызов подпрограммы: SUB_PROG
N60 ...	
...	
N100 M30	; конец программы

Подпрограмма:

Программный код	Комментарий
PROC SUB_PROG	
...	
N100 RET	; Возврат выполняется на кадр N60 в главной программе.

2.24.2.10 Параметризуемый возврат из подпрограммы (RET ...)

Вообще с помощью команды `RET` выполняется возврат из подпрограммы в вызывающую программу. Затем обработка продолжается с программной строки, следующей за вызовом подпрограммы. Если обработка программы должна продолжаться с другого места, имеются следующие возможности:

- Продолжение обработки программы после вызова циклов обработки резаньем в режиме диалекта ISO (после описания контура).
- Возврат в главную программу из любого уровня подпрограммы (и после ASUP) при обработке ошибок.
- Возврат через несколько программных уровней для специальных приложений в компилируемых циклах и в режиме диалекта ISO.

Для этого команду `RET` нужно запрограммировать с дополнительными параметрами.

Направление поиска

При указании параметра <конечный кадр> возврат осуществляется сначала на кадр после кадра вызова. Затем выполняется поиск цели в направлении **конца** программы, в которую осуществлен возврат. Если этот поиск не дал результата, далее выполняется поиск в направлении начала программы.

Синтаксис

```
RET ("<конечный кадр>")
RET ("<конечный кадр>",<кадр после конечного кадра>)
RET ("<конечный кадр>",<кадр после конечного кадра>,<число уровней возврата>)
RET ("<конечный кадр>",<число уровней возврата>)
RET ("<конечный кадр>",<кадр после конечного кадра>,<число уровней возврата>,<возврат на начало программы>)
```

RET (, , <число уровней возврата>, <возврат на начало программы>)

Объяснение

RET:	Конец подпрограммы		
<конечный кадр>:	В этом параметре в качестве цели перехода указывается кадр, на котором обработка программы должна быть продолжена. Если параметр <число уровней возврата> не запрограммирован, то цель перехода находится в программе, из которой была вызвана актуальная подпрограмма. Возможными данными являются:		
	<номер кадра>	Номер конечного кадра. Поиск номера кадра в программе, куда выполняется возврат, сначала осуществляется в направлении конца программы.	
	<метка перехода>	Метка перехода, которая должна быть в программе, куда выполняется возврат. Поиск метки перехода в программе, куда выполняется возврат, сначала осуществляется в направлении конца программы.	
	<строка символов>	Строка символов, которая должна быть в программе, куда выполняется возврат (например, имя программы или переменной). Поиск строки символов в программе, куда выполняется возврат, сначала осуществляется в направлении конца программы. Для программирования строки символов в конечном кадре действуют следующие правила: • Пробел на конце (в отличие от метки перехода, которая обозначается через ":" на конце). • Перед строкой символов может быть установлен только номер кадра и/или метка перехода, не программные команды.	
<кадр после конечного кадра>:	В этом параметре указывается, должна ли обработка программы продолжаться в кадре, указанном в параметре <конечный кадр>, или в следующем за ним кадре.		
	Тип:	INT	
	Значение:	0	Возврат осуществляется на кадр, указанный в параметре <конечный кадр>.
> 0		Возврат осуществляется на кадр, следующий за указанным в параметре <конечный кадр>.	

<число уровней возврата>:	В этом параметре указывается число программных уровней, которые нужно пройти назад, чтобы найти там конечный кадр и продолжить обработку программы.	
	Тип:	INT
	Значение:	1 Программы будет продолжена на "актуальном программном уровне - 1" (как RET без параметров).
		2 Программы будет продолжена на "актуальном программном уровне - 2", т. е. будет пропущен один уровень.
		3 Программы будет продолжена на "актуальном программном уровне - 3", т. е. будет пропущено два уровня.
<возврат на начало программы>:	...	
	Диапазон значений:	1 ... 15
<возврат на начало программы>:	В этом параметре указывается, нужно ли при возврате в главную программу и активном там режиме диалекта ISO продолжать программу с самого начала.	
	Тип:	BOOL
	Значение:	1 Если возврат осуществляется в главную программу и там активен режим диалекта ISO , то выполняется ветвление на начало программы.

Примечание

При возврате из подпрограммы со строкой символов в качестве данных для поиска конечного кадра, то в вызывающей программе сначала всегда выполняется поиск метки перехода.

Поэтому, если цель перехода должна быть однозначно определена строкой символов, то строка символов не должна совпадать с именем метки перехода, т.к. в ином случае возврат из подпрограммы всегда будет выполняться на метку перехода, а не на строку символов (см. пример 2).

Граничные условия

При возврате через несколько программных уровней обрабатываются операторы SAVE отдельных программных уровней.

Если при возврате через несколько программных уровней активна модальная подпрограмма и если в одной из пропущенных подпрограмм запрограммирована

команда отмены MCALL для модальной подпрограммы, то модальная подпрограмма остается активной.

ВНИМАНИЕ

Ошибки программирования

В случае возврата через несколько программных уровней пользователь сам должен позаботиться о том, чтобы обработка продолжалась с необходимыми модальными настройками. Это обеспечивается, например, программированием соответствующего главного кадра.

Примеры

Пример 1: Повторная установка в главной программе после обработки ASUP

Программирование	Комментарий
N10010 CALL "UP1"	; программный уровень 0 (главная программа)
N11000 PROC UP1	; программный уровень 1
N11010 CALL "UP2"	
N12000 PROC UP2	; программный уровень 2
...	
N19000 PROC ASUP	; программный уровень 3 (обработка ASUP)
...	
N19100 RET("N10900", , \$P_STACK)	; Возврат из подпрограммы в главную программу
	; \$P_STACK: актуальный программный уровень
N10900	; Конечный кадр в главной программе
N10910 MCALL	; Выключить модальный вызов подпрограммы
N10920 G0 G60 G40 M5	; Инициализировать другие модальные настройки

Пример 2: Строка символов (<строка>) как данные для поиска конечного кадра

Главная программа:

Программный код	Комментарий
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; Вызов подпрограммы "subProg1"
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; Вызов подпрограммы "subProg2"
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1: iVar1=R10*44	
N1510 F500 X5	

Программный код	Комментарий
N1520 ...	
N1550 subprog1: G1 X30	; "subProg1" определена здесь как метка перехода.
N1560 ...	
N1600 subProg3	; Вызов подпрограммы "subProg3"
N1610 ...	
N1900 M30	

Подпрограмма subProg1:

Программный код	Комментарий
PROC subProg1	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; Возврат в главную программу на кадр N1400

Подпрограмма subProg2:

Программный код	Комментарий
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; Возврат в главную программу на кадр N1500

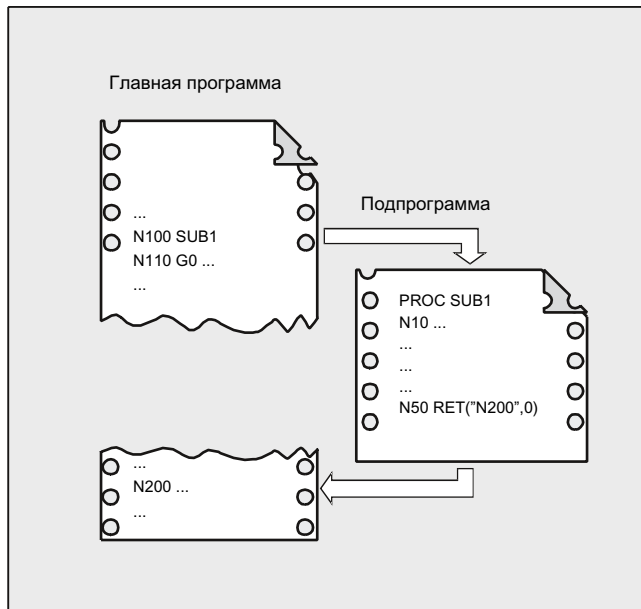
Подпрограмма subProg3:

Программный код	Комментарий
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; Возврат в главную программу на кадр N1550

Дополнительная информация

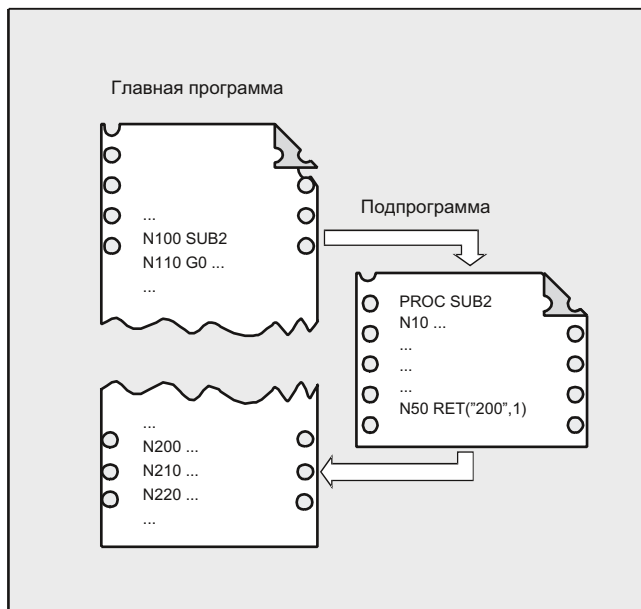
На следующих рисунках показаны различные действия параметров возврата

1. <конечный кадр> = "N200", <кадр после конечного кадра> = 0



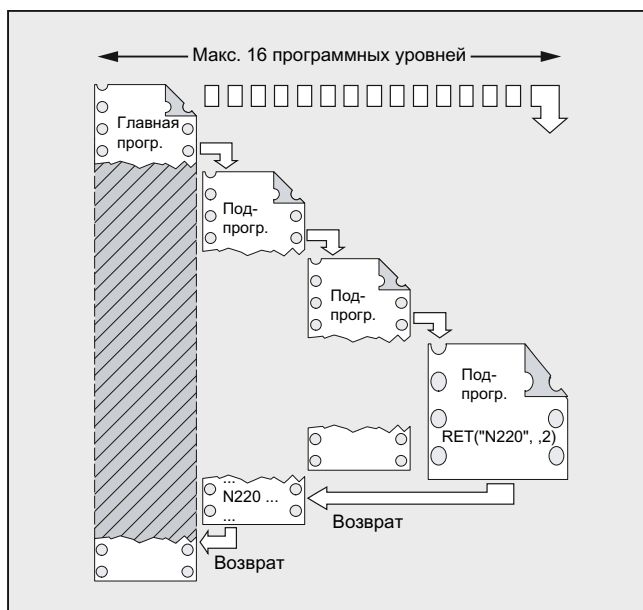
После команды `RET` обработка программы продолжается на кадре N200 в главной программе.

2. <конечный кадр> = "N200", <кадр после конечного кадра> = 1



После команды `RET` обработка программы продолжается на кадре (N210), следующим за кадром N200 в главной программе.

3. <конечный кадр> = "N220", <число уровней возврата> = 2



После команды `RET` два программных уровня пропускается и обработка программы продолжается на кадре `N220`.

2.24.2.11 Параметризуемый возврат из подпрограммы (`RETB ...`)

Вообще с помощью команды `RETB` выполняется возврат из подпрограммы в вызывающую программу. Затем обработка продолжается с программной строки, следующей за вызовом подпрограммы. Если обработка программы должна продолжаться с другого места, имеются следующие возможности:

- Продолжение обработки программы после вызова циклов обработки резаньем в режиме диалекта ISO (после описания контура).
- Возврат в главную программу из любого уровня подпрограммы (и после `ASUP`) при обработке ошибок.
- Возврат через несколько программных уровней для специальных приложений в компилируемых циклах и в режиме диалекта ISO.

Для этого команду `RETB` нужно запрограммировать с дополнительными параметрами.

Направление поиска

При указании параметра <конечный кадр> возврат осуществляется сначала на кадр после кадра вызова. Затем выполняется поиск цели в направлении **начала** программы, в которую осуществлен возврат. Если этот поиск не дал результата, далее выполняется поиск в направлении конца программы.

Синтаксис

```
RETB ("<конечный кадр>")
RETB ("<конечный кадр>",<кадр после конечного кадра>)
RETB ("<конечный кадр>",<кадр после конечного кадра>,<число уровней возврата>)
```

RETB ("<конечный кадр>", , <число уровней возврата>)

RETB ("<конечный кадр>", <кадр после конечного кадра>, <число уровней возврата>, <возврат на начало программы>)

RETB (, , <число уровней возврата>, <возврат на начало программы>)

Объяснение

RETB:	Конец подпрограммы		
<конечный кадр>:	В этом параметре в качестве цели перехода указывается кадр, на котором обработка программы должна быть продолжена. Если параметр <число уровней возврата> не запрограммирован, то цель перехода находится в программе, из которой была вызвана актуальная подпрограмма. Возможными данными являются:		
	<номер кадра>	Номер конечного кадра. Поиск номера кадра в программе, куда выполняется возврат, сначала осуществляется в направлении начала программы.	
	<метка перехода>	Метка перехода, которая должна быть в программе, куда выполняется возврат. Поиск метки перехода в программе, куда выполняется возврат, сначала осуществляется в направлении начала программы.	
	<строка символов>	Строка символов, которая должна быть в программе, куда выполняется возврат (например, имя программы или переменной). Поиск строки символов в программе, куда выполняется возврат, сначала осуществляется в направлении начала программы. Для программирования строки символов в конечном кадре действуют следующие правила: • Пробел на конце (в отличие от метки перехода, которая обозначается через ":" на конце). • Перед строкой символов может быть установлен только номер кадра и/или метка перехода, не программные команды.	
	<кадр после конечного кадра>:	В этом параметре указывается, должна ли обработка программы продолжаться в кадре, указанном в параметре <конечный кадр>, или в следующем за ним кадре.	
	Тип:	INT	
Значение:	0	Возврат осуществляется на кадр, указанный в параметре <конечный кадр>.	
	> 0	Возврат осуществляется на кадр, следующий за указанным в параметре <конечный кадр>.	

<число уровней возврата>:	В этом параметре указывается число программных уровней, которые нужно пройти назад, чтобы найти там конечный кадр и продолжить обработку программы.	
	Тип:	INT
	Значение:	1 Программы будет продолжена на "актуальном программном уровне - 1" (как RET без параметров).
		2 Программы будет продолжена на "актуальном программном уровне - 2", т. е. будет пропущен один уровень.
		3 Программы будет продолжена на "актуальном программном уровне - 3", т. е. будет пропущено два уровня.
<возврат на начало программы>:	...	
	Диапазон значений:	1 ... 15
<возврат на начало программы>:	В этом параметре указывается, нужно ли при возврате в главную программу и активном там режиме диалекта ISO продолжать программу с самого начала.	
	Тип:	BOOL
	Значение:	1 Если возврат осуществляется в главную программу и там активен режим диалекта ISO , то выполняется ветвление на начало программы.

Примечание

При возврате из подпрограммы со строкой символов в качестве данных для поиска конечного кадра, то в вызывающей программе сначала всегда выполняется поиск метки перехода.

Поэтому, если цель перехода должна быть однозначно определена строкой символов, то строка символов не должна совпадать с именем метки перехода, т.к. в ином случае возврат из подпрограммы всегда будет выполняться на метку перехода, а не на строку символов (см. пример 2).

Граничные условия

При возврате через несколько программных уровней обрабатываются операторы SAVE отдельных программных уровней.

Если при возврате через несколько программных уровней активна модальная подпрограмма и если в одной из пропущенных подпрограмм запрограммирована

ВНИМАНИЕ

В случае возврата через несколько программных уровней пользователь сам должен позаботиться о том, чтобы обработка продолжалась с необходимыми модальными настройками. Это обеспечивается, например, программированием соответствующего главного кадра.

Пример

Программный код	Комментарий
BEISPIEL.MPF	
...	
N3000 START_CYC(param1, param2, ...)	
N3010 TECH_CYC1(param1, param2, ...)	
N3020 TECH_CYC2(param1, param2, ...)	
N3030 TECH_CYC3(param1, param2, ...)	
N3040 END_CYC(param1, param2, ...)	
N3040 END_CYC (param1, param2, ...)	
N3050 ...	
N4500 START_CYC(param11, param12, ...)	
N4510 ...	
N4590 END_CYC(param11, param12, ..)	
N5000 ...	
...	
N6000 M30	

Программный код	Комментарий
PROC END_CYC(...)	; Вызов в главной программе строки N3040
N10000 ...	
N15000 if status == 1	
N15010 RETB("START_CYC")	; Возврат в вызывающую программу BEISPIEL.MPF
	; Поиск строки символов "START_CYC"
	; Направление поиска: назад, в направлении начала программы
	; Продолжение обработки программы со строки N3000
N15020 ENDIF	
N15030 if status == 0	
N15040 RET	; Возврат в вызывающую программу BEISPIEL.MPF
	; Продолжение обработки программы со строки N3050
N15050 ENDIF	

Программный код	Комментарий
N16000 RET("START_CYC")	; Возврат в вызывающую программу BEISPIEL.MPF ; Поиск строки символов "START_CYC" ; Направление поиска: вперед, в направлении конца программы ; Продолжение обработки программы со строки N4500
N17060 RETB	; Возврат в вызывающую программу BEISPIEL.MPF ; Продолжение обработки программы со строки N3050 ; RETB без параметров идентична RET

2.24.3 Вызов подпрограммы

2.24.3.1 Вызовы подпрограмм без передачи параметров

Вызов подпрограммы выполняется либо с адресом L и номером подпрограммы, либо через указание имени программы.

Главная программа также может быть вызвана как подпрограмма. В этом случае установленный в главной программе конец программы M2 или M30 интерпретируется как M17 (конец программы с возвратом в вызывающую программу).

Примечание

Подпрограмма также может быть запущена как главная программа.

Стратегия поиска СЧПУ:

Существует ли *_MPF?

Существует ли *_SPF?

Из этого следует: Если имя вызываемой подпрограммы идентично имени главной программы, то снова вызывается вызывающая главная программа. Этого, как правило, нежелательного эффекта можно избежать через выбор однозначных имен для подпрограмм и главных программ.

Примечание

Подпрограммы, не требующие передачи параметров, могут вызываться и из файла инициализации.

Синтаксис

L<номер>/<имя программы>

Примечание

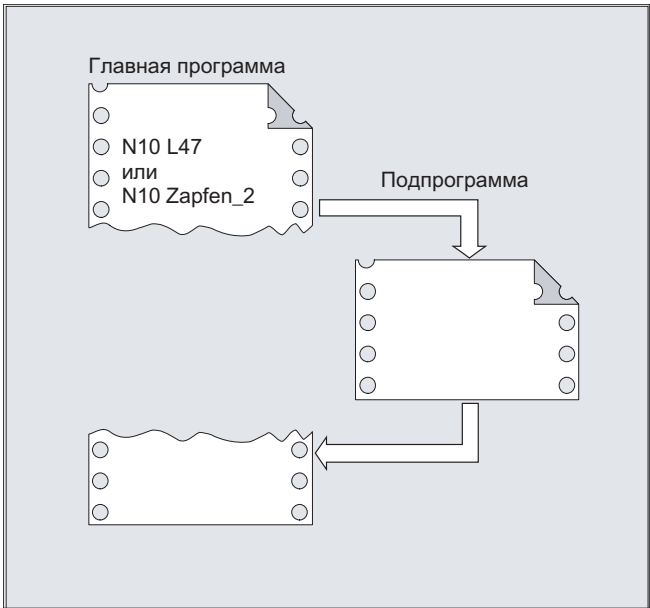
Вызов подпрограммы всегда должен быть запрограммирован в отдельном кадре ЧПУ.

Значение

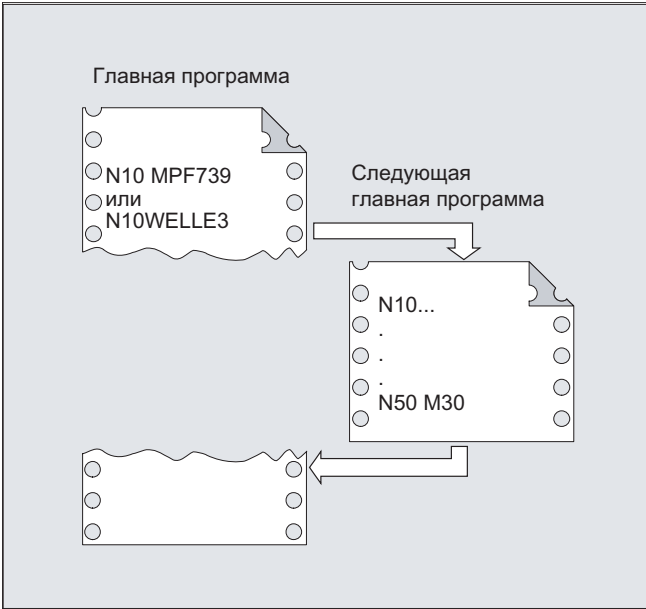
L:	Адрес для вызова подпрограммы	
<номер>:	Номер подпрограммы	
	Тип:	INT
	Значение:	Макс. 7 десятичных позиций Внимание: Вводные нули имеют значение при присваивании имени (⇒ L123, L0123 и L00123 это три различные подпрограммы).
<имя программы>:	Имя подпрограммы (или главной программы)	

Примеры

Пример 1: Вызов подпрограммы без передачи параметров



Пример 2: Вызов главной программы как подпрограммы



См. также

Подпрограмма без передачи параметров (Страница 171)

2.24.3.2 Вызов подпрограммы с передачей параметров (EXTERN)

При вызове подпрограммы с передачей параметров переменные или значения могут передаваться напрямую (не для параметров VAR).

Подпрограммы с передачей параметров перед вызовом в главной программе должны быть указаны с EXTERN (к примеру, в начале программы). При этом указывается имя подпрограммы и типы переменных в порядке передачи.

ВНИМАНИЕ
Опасность спутывания
Как типы переменных, так и последовательность передачи, должны совпадать с определениями, согласованными в подпрограмме в PROC. Имена параметров в главной и подпрограмме могут быть различными.

Синтаксис

```
EXTERN <имя программы>(<тип_пар1>,<тип_пар2>,<тип_пар3>)  
...  
<имя программы>(<значение_пар1>,<значение_пар2>,<значение_пар3>)
```

Примечание

Вызов подпрограммы всегда должен быть запрограммирован в отдельном кадре ЧПУ.

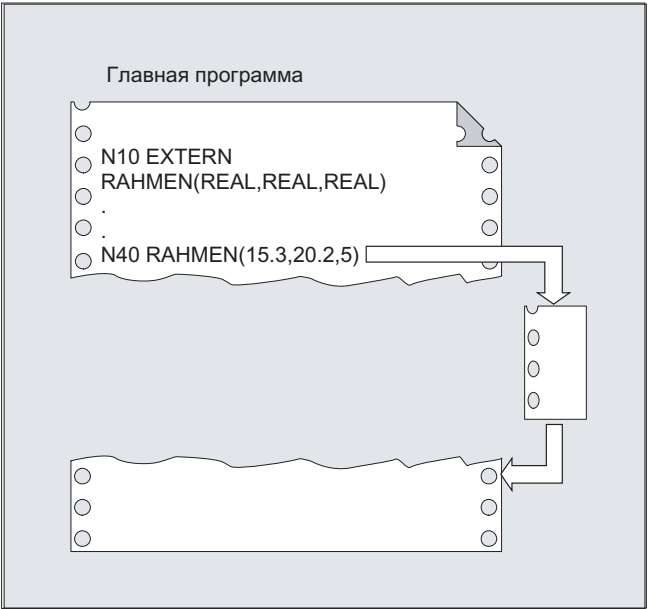
Объяснение

<имя программы>:	Имя подпрограммы
EXTERN:	Кодовое слово для указания подпрограммы с передачей параметров Примечание: EXTERN должен указываться только тогда, когда подпрограмма стоит в директории детали или в глобальной директории подпрограмм. Циклы не должны указываться как EXTERN .
<тип_пар1>,<тип_пар2>,<тип_пар3>:	Типы переменных передаваемых параметров в последовательности передачи
<значение_пар1>,<значение_пар2>,<значение_пар3>:	Значения переменных для передаваемых параметров

Примеры

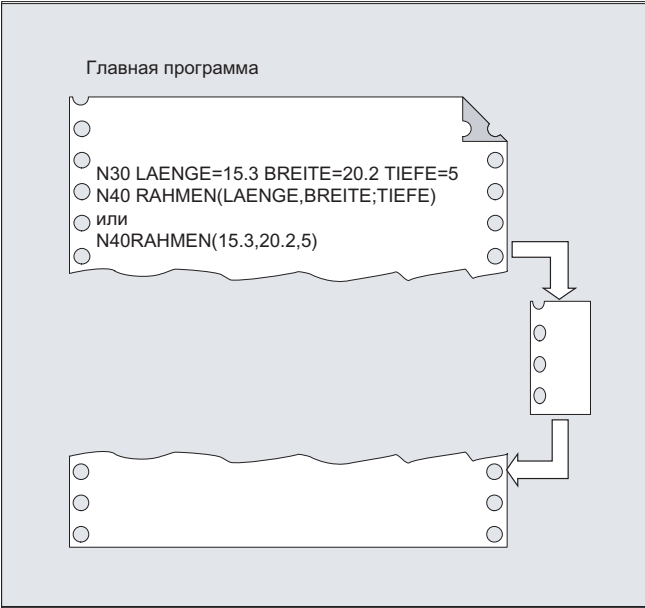
Пример 1: Вызов подпрограммы с предшествующим извещением

Программный код	Комментарий
N10 EXTERN RAHMEN (REAL, REAL, REAL)	; Указание подпрограммы.
...	
N40 RAHMEN (15.3, 20.2, 5)	; Вызов подпрограммы с передачей параметров.



Пример 2: Вызов подпрограммы без извещения

Программный код	Комментарий
N10 DEF REAL LAENGE, BREITE, TIEFE	
N20 ...	
N30 LAENGE=15.3 BREITE=20.2 TIEFE=5	
N40 RAHMEN (LAENGE,BREITE,TIEFE)	; или: N40 RAHMEN (15.3,20.2,5)



См. также

Подпрограмма с передачей параметров по значению (PROC) (Страница 172)

Подпрограмма с передачей параметров по ссылке (PROC, VAR) (Страница 174)

2.24.3.3 Число повторений программы (P)

Если необходимо последовательное выполнение подпрограммы несколько раз, то в кадре с вызовом подпрограммы по адресу P можно запрограммировать желаемое число повторений программы.

 **ОСТОРОЖНО**

Вызов подпрограммы с повторением программы и передачей параметров
Параметры передаются только при вызове или первом проходе программы. Для следующих повторений параметры остаются неизменными. Если необходимо при повторениях программы изменить параметры, то следует установить в подпрограмме соответствующие согласования.

Синтаксис

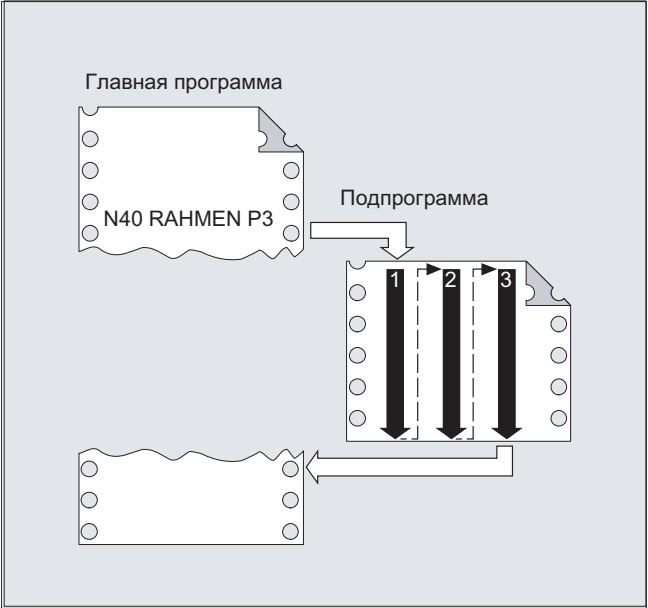
<имя программы> P<значение>

Значение

<имя программы>:	Вызов подпрограммы	
P:	Адрес для программирования повторений программы	
<значение>:	Число повторений программы	
	Тип:	INT
	Диапазон значений:	1 ... 9999 (без знака)

Пример

Программный код	Комментарий
...	
N40 RAHMEN P3	; Подпрограмма RAHMEN должна быть выполнена последовательно три раза.
...	



2.24.3.4 Модальный вызов подпрограммы (MCALL)

При модальном вызове подпрограммы `MCALL` (<имя программы>) указанная подпрограмма вызывается не сразу. Вместо этого вызов с этого момента в программе обработки детали осуществляется автоматически после каждого кадра перемещения с движениями по траектории. В том числе и с переходом через программные уровни.

Примечание

При выполнении программы действует всегда только последний модальный вызов подпрограммы `MCALL` (<имя программы>). Текущий модальный вызов подпрограммы заменяет активный прежде.

Если в подпрограмму передаются параметры, то передача параметров осуществляется только при вызове `MCALL` (<имя программы>(`Par1`, `Par2`, ...)).

ВНИМАНИЕ

Модальные вызовы подпрограммы без движения по траектории

Модальная подпрограмма в следующих ситуациях вызывается и без программирования движения по траектории:

- Программирование адресов `S` или `F`, если активна `G0` или `G1`
- Если `G0` или `G1` стоит в кадре одна или была запрограммирована вместе с другими `G`-командами.

Синтаксис

```
MCALL <имя программы>
...
MCALL
```

Объяснение

<code>MCALL</code> <имя программы>:	Включение функции "Модальный вызов подпрограммы"
<имя программы>:	Имя подпрограммы
<code>MCALL</code> :	С помощью <code>MCALL</code> без указания имени программы функция "Модальный вызов подпрограммы" выключается.

Граничные условия

ASUP

Если обработка программы обработки детали прерывается программой `ASUP` (см. главу "Обработчик прерываний (`ASUP`) (Страница 134)"), то в этой `ASUP` модальные вызовы подпрограммы не выполняются.

Если ASUP запускается в состоянии канала "Reset", она ведет себя относительно модальных вызовов подпрограммы, как обычная программа обработки детали.

Цикл смены инструмента

Если в цикле смены инструмента происходит отмена функции "Модальный вызов подпрограммы", следует учитывать, что при необходимости цикл смены инструмента не явно вызывается и после поиска кадра через поисковую ASUP или вручную через пересохранение. В этой ситуации отменять функцию "Модальный вызов подпрограммы" не следует, потому что иначе результат поиска будет ошибочным. Поэтому отмену функции "Модальный вызов подпрограммы" в цикле смены инструмента рекомендуется программировать следующим образом:

Программный код	Комментарий
...	
IF \$AC_ASUP == 0	; Вызов осуществляется не через поисковую ASUP или пересохранение.
MCALL	; Выключение функции "Модальный вызов подпрограммы".
ENDIF	
...	

Примеры

Пример 1

Программный код	Комментарий
N10 G0 X0 Y0	
N20 MCALL L70	; Включение модального вызова подпрограммы для L70.
N30 X10 Y10	; Выполняется подвод в X10 Y10, затем вызывается L70.
N40 X20 Y20	; Выполняется подвод в X20 Y20, затем вызывается L70.
...	
N100 MCALL	; Выключение функции "Модальный вызов подпрограммы".
N110 X0 Y0	; Выполняется подвод в X0 Y0, а L70 не вызывается.

Пример 2

Программный код
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

В этом примере последующие кадры ЧПУ с запрограммированными траекторными осями стоят в подпрограмме L80. L70 вызывается через L80.

2.24.3.5 Косвенный вызов подпрограммы (CALL)

В зависимости от имеющихся условий в одном месте могут вызываться различные подпрограммы. Для этого имя подпрограммы сохраняется в переменной типа STRING. Вызов подпрограммы осуществляется с CALL и именем переменной.

Примечание

Косвенный вызов подпрограммы возможен только для подпрограмм без передачи параметров. Для прямого вызова подпрограммы сохранить имя в постоянной STRING.

Синтаксис

CALL <имя программы>

Объяснение

CALL:	Команда для косвенного вызова подпрограммы	
<имя программы>:	Имя подпрограммы (переменная или постоянная)	
	Тип:	STRING

Пример

Прямой вызов с постоянной STRING:

Программный код	Комментарий
...	
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"	; Вызвать подпрограмму TEIL1 с CALL напрямую.
...	

Косвенный вызов через переменную:

Программный код	Комментарий
...	
DEF STRING[100] PROGNAME	; Определение переменных.
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_TEIL1_SPF"	; Согласовать подпрограмму TEIL1 с переменной PROGNAME.
CALL PROGNAME	; Вызвать подпрограмму TEIL1 через CALL и переменную PROGNAME косвенно.
...	

2.24.3.6 Косвенный вызов подпрограммы с указанием выполняемого программного блока (CALL BLOCK ... TO ...)

С помощью `CALL` и комбинации кодовых слов `BLOCK ... TO` подпрограмма вызывается косвенно и выполняется обозначенный стартовой меткой и меткой конца программный блок.

Синтаксис

```
CALL <имя программы> BLOCK <стартовая метка> TO <метка конца>
CALL BLOCK <стартовая метка> TO <метка конца>
```

Значение

CALL:	Команда для косвенного вызова подпрограммы	
<имя программы>:	Имя подпрограммы (переменная или постоянная), содержащей обрабатываемый программный блок (опциональные данные).	
	Тип:	STRING
	Примечание: Если <имя программы> не запрограммировано, то поиск и выполнение обозначенного <стартовой меткой> и <меткой конца> программного блока осуществляется в актуальной программе.	
BLOCK ... TO ... :	Комбинация кодовых слов для косвенного выполнения программного блока	
<стартовая метка>:	Переменная, указывающая на начало обрабатываемого программного блока.	
	Тип:	STRING
<метка конца>:	Переменная, указывающая на конец обрабатываемого программного блока.	
	Тип:	STRING

Пример

Главная программа:

Программный код	Комментарий
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; Определение переменных для стартовой метки и метки конца.
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL	; Косвенный вызов подпрограммы и обозначение выполняемого программного блока.
...	

Подпрограмма:

Программный код	Комментарий
PROC CONTUR_1 ...	
LABEL_1	; Стартовая метка: Начало выполнения программного блока
N1000 G1 ...	
...	
LABEL_2	; Метка конца: Конец выполнения программного блока
...	

2.24.3.7 Косвенный вызов запрограммированной на языке ISO программы (ISOCALL)

С помощью косвенного вызова программы `ISOCALL` можно вызвать запрограммированную на языке ISO программу. При этом активируется установленный в машинных данных режим ISO. В конце программы снова активируется первоначальный режим обработки. Если в машинных данных режим ISO не установлен, то вызов подпрограммы выполняется в режиме Siemens.

Прочую информацию по режиму ISO см.:

Литература:

Описание функций диалекта ISO

Синтаксис

`ISOCALL <имя программы>`

Объяснение

<code>ISOCALL:</code>	Кодовое слово для косвенного вызова подпрограммы, при котором активируется установленный в машинных данных режим ISO.
<code><имя программы>:</code>	Имя запрограммированной на языке ISO программы (переменная или постоянная типа STRING)

Пример: Пример вызова контура с программированием циклов из режима ISO

Программный код	Комментарий
0122_SPF	; Описание контура в режиме ISO
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] PROGNAME = "0122"	; Программа (цикл) обработки детали Siemens
...	
N2000 R11 = \$AA_IW[X]	
N2010 ISOCALL PROGNAME	
N2020 R10 = R10+1	; Выполнить программу 0122.spf в режиме ISO

Программный код	Комментарий
...	
N2400 M30	

2.24.3.8 Вызов подпрограммы с указанием пути и параметрами (PCALL)

С помощью `PCALL` возможен вызов подпрограмм с абсолютным указанием пути и передачей параметров.

Синтаксис

`PCALL <путь/имя программы>(<параметр 1>,..., <параметр n>)`

Значение

<code>PCALL:</code>	Кодовое слово для вызова подпрограммы с абсолютным указанием пути.
<code><путь/имя программы>:</code>	Абсолютное указание пути включая имя подпрограммы. Правила указания пути см. "Адресация файлов программной памяти (Страница 227)". Если абсолютный путь не был указан, то поведение <code>PCALL</code> идентично стандартному вызову подпрограммы с идентификатором программы. Имя программы указывается без префикса и без идентификатора файла. Если имя программы нужно запрограммировать с префиксом и идентификатором файла, то оно должно быть явно интерпретировано с префиксом и идентификатором файла с помощью команды <code>EXTERN</code> .
<code><параметр 1>, ...,:</code>	Актуальные параметры согласно оператору <code>PROC</code> подпрограммы.

Пример

Программный код
<code>PCALL/_N_WKS_DIR/_N_WELLE_WPD/WELLE (параметр1, параметр2,...)</code>

2.24.3.9 Расширить маршрут поиска для вызовов подпрограмм (CALLPATH)

С помощью команды `CALLPATH` можно расширить маршрут поиска для вызовов подпрограммы. Таким образом, могут вызываться и подпрограммы из не выбранной директории детали без указания полного, абсолютного имени пути подпрограммы.

Другой способ применения предлагается в режиме работы EES "EES без GDIR", когда для хранения глобальных подпрограмм используется директория во внешней программной памяти. В этом случае с помощью `CALLPATH` можно расширить маршрут поиска, включив в него эту директорию подпрограммы.

Расширение маршрута поиска выполняется перед строкой для циклов пользователя (`_N_CUS-DIR`).

Следующие события снова отменяют расширение маршрута поиска:

- CALLPATH с пробелом
- CALLPATH без параметров
- Конец программы обработки детали
- Reset

Синтаксис

CALLPATH ("*<имя маршрута>*")

Объяснение

CALLPATH:	Кодовое слово для программируемого расширения маршрута поиска. Программируется в собственной строке программы обработки детали.
<i><имя маршрута></i> :	Постоянная или переменная типа STRING. Содержит абсолютное указание пути к директории, на которую должен быть расширен маршрут поиска. Правила указания пути см. "Адресация файлов программной памяти (Страница 227)".

Пример

Маршрут поиска нужно расширить на определенную директорию детали:

Программный код
...
CALLPATH ("/_N_WKS_DIR/_N_MYWPD_WPD")
...

Здесь устанавливается следующий маршрут поиска (позиция 5 является новой):

1. Актуальная директория/*name*
2. Актуальная директория/*name_SPF*
3. Актуальная директория/*name_MPF*
4. //NC:/_N_SPF_DIR/*name_SPF*
5. /_N_WKS_DIR/_N_MYWPD_WPD/*name_SPF*
6. /N_CUS_DIR/*name_SPF*
7. /_N_CMA_DIR / *name_SPF*
8. /_N_CST_DIR / *name_SPF*

Граничные условия

- `CALLPATH` проверяет, существует ли запрограммированное имя маршрута в действительности. В случае ошибки выполнение программы обработки детали отменяется с ошибкой кадра коррекции 14009.
- `CALLPATH` может быть запрограммирована и в файлах INI. В этом случае он действует в течение срока обработки файла INI (файл WPD-INI или программа инициализации для активных данных ЧПУ, к примеру, фреймы в 1-ом канале `_N_CH1_UFR_INI`). После маршрут поиска снова сбрасывается.

2.24.3.10 Выполнение внешней подпрограммы (840D sl) (EXTCALL)

С помощью команды `EXTCALL` можно догрузить и выполнить программу обработки детали из внешнего ЗУ.

В качестве внешнего ЗУ предлагаются:

- Локальный диск
- Сетевой диск
- Диск USB

Примечание

В качестве интерфейса для обработки находящейся на USB-диске внешней программы можно использовать только интерфейсы USB на лицевой стороне пульта управления или TCU.

ВНИМАНИЕ**Повреждение инструмента/детали при использовании USB-флэшки**

Не рекомендуется использовать USB-флэшку для обработки внешней подпрограммы. Нарушение коммуникации с USB-флэшкой при выполнении программы обработки детали из-за плохого контакта, толчков или случайного извлечения приводит к немедленной остановке обработки. При этом инструмент и/или деталь могут быть повреждены.

Предустановка ветви программы

Путь к внешней директории программы может быть предустановлен с помощью установочных данных:

SD42700 \$SC_EXT_PROG_PATH

В комбинации с указанной при вызове `EXTCALL` ветвью программы и идентификатором подпрограммы из этого получается общий путь к вызываемой программе обработки детали.

Примечание**Параметр**

При вызове внешней программы передача в нее параметров невозможна.

Синтаксис

```
EXTCALL ("<путь/><имя программы>")
```

Объяснение

EXTCALL:	Команда для вызова внешней подпрограммы	
"<путь/><имя программы>":	Постоянная/переменная типа STRING	
	<путь/>:	Абсолютное или относительное указание пути (опция)
	<имя программы>:	Имя программы указывается без префикса "_N_". Расширение файла ("MPF", "SPF") может быть добавлено с символом "_" или "." к имени программы (опция). Пример: "WELLE" "WELLE_SPF" "WELLE.SPF"

Указание пути: Краткие обозначения

При указании пути можно использовать следующие условные обозначения:

- Локальный диск: "LOCAL_DRIVE:"
- CF-карта: "CF_CARD:"
- USB-диск (панель оператора): "USB:"

Краткие обозначения "CF_CARD:" и "LOCAL_DRIVE:" можно использовать в качестве альтернативы.

Пример**Выполнение с локального диска**

Главная программа "MAIN.MPF" находится в памяти ЧПУ и выбрана для выполнения.

Подпрограмма "SP_1"

Внешняя подпрограмма "SP_1.SPF" или "SP_1.MPF" находится на локальном диске в директории "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

Путь к внешней директории программ устанавливается с:

```
SD42700 $SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD
```

Примечание

Указание пути для вызова внешней подпрограммы:

- Без использования предустановки: "LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
- С использованием предустановки: "SP_1"

Подпрограмма "SP_2"

Внешняя подпрограмма "SP_2.SPF" или "SP_2.MPF" находится в директории WKS.DIR/ WST1.WPD диска USB. Предустановка пути к внешней директории программ используется для пути подпрограммы "SP_1" и также не переписывается в главной программе. Поэтому при вызове подпрограммы "SP_2" должен быть указан полный путь.

Главная программа "MAIN"**Программный код**

```
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

Дополнительная информация**Вызов EXTCALL с абсолютным указанием пути**

Если подпрограмма существует по указанному пути, то она выполняется с вызовом EXTCALL. Если подпрограмма отсутствует по указанному пути, то выполнение программы отменяется после вызова EXTCALL.

Вызов EXTCALL с относительным указанием пути / без указания пути

При вызове EXTCALL с относительным указанием пути или без указания пути поиск в имеющейся программной памяти осуществляется по следующему образцу:

1. Если в SD42700 \$SC_EXT_PROG_PATH предустановленно указание пути, то сначала, исходя из этого пути, выполняется поиск данных в вызове EXTCALL (имя программы возможно с относительным указанием пути). В этом случае абсолютный путь получается через соединение символов:
 - Предустановленное указание пути в SD42700 \$SC_EXT_PROG_PATH
 - Разделительный символ "/"
 - Указание пути и имя подпрограммы в команде EXTCALL
2. Если подпрограмма не была найдена в 1, то поиск выполняется в директориях памяти пользователя.

Поиск прекращается при первом нахождении подпрограммы. Если подпрограмма не найдена, то выполнение программы отменяется после вызова EXTCALL.

Настраиваемая память догрузки (буфер FIFO)

Для выполнения внешней подпрограммы потребуется память догрузки. Размер памяти догрузки предустановлен на 30 кБ и может быть изменен только изготовителем станка (через MD18360 MM_EXT_PROG_BUFFER_SIZE).

Примечание**Подпрограммы с операторами переходов**

Для внешних подпрограмм, содержащих операторы перехода (GOTO, GOTO, CASE, FOR, LOOP, WHILE, REPEAT, IF, ELSE, ENDIF и т. п.), цели перехода должны находиться в памяти догрузки.

Примечание**Программы ShopMill-/ShopTurn**

Программы ShopMill и ShopTurn из-за прикрепленных на конце файла описаний контура должны целиком лежать в памяти догрузки.

Для выполняемых параллельно внешних подпрограмм требуется своя память догрузки.

Reset / конец программы / POWER ON

RESET и POWER ON отменяют внешние вызовы подпрограмм и соответствующая память догрузки стирается.

Выбранная для "Выполнения с внешнего устройства" подпрограмма остается выбранной для "Выполнения с внешнего устройства" и после Reset, завершения программы или POWER ON. Поведение не отличается от внутренне выбранных программ, если внешняя программная память остается доступной.

Литература

Дополнительную информацию по "Выполнению с внешнего устройства" можно найти в: Описание функций "Основные функции"; GPP, канал, программный режим, параметры Reset (K1)

2.24.3.11 Выполнение внешней подпрограммы (828D) (EXTCALL)

С помощью команды EXTCALL можно догрузить и выполнить программу обработки детали из внешнего ЗУ.

В качестве внешнего ЗУ предлагаются:

- CF-карта пользователя
- Сетевой диск
- Диск USB

Примечание

В качестве интерфейса для обработки находящейся на диске USB внешней программы можно использовать только интерфейс USB панели оператора (PPU).

ВНИМАНИЕ**Повреждение инструмента/детали при использовании USB-флэшки**

Не рекомендуется использовать USB-флэшку для обработки внешней подпрограммы. Нарушение коммуникации с USB-флэшкой при выполнении программы обработки детали из-за плохого контакта, толчков или случайного извлечения приводит к немедленной остановке обработки. При этом инструмент и/или деталь могут быть повреждены.

Предустановка ветви программы

Путь к внешней директории программы может быть предустановлен с помощью установочных данных:

SD42700 \$SC_EXT_PROG_PATH

В комбинации с указанной при вызове `EXTCALL` ветвью программы и идентификатором подпрограммы из этого получается общий путь к вызываемой программе обработки детали.

Примечание**Параметр**

При вызове внешней программы передача в нее параметров невозможна.

Синтаксис

`EXTCALL ("<путь/><имя программы>")`

Объяснение

EXTCALL:	Команда для вызова внешней подпрограммы	
<"<путь/><имя программы>":	Постоянная/переменная типа STRING	
	<путь/>:	Абсолютное или относительное указание пути (опция)
	<имя программы>:	Имя программы указывается без префикса "_N_". Расширение файла ("MPF", "SPF") может быть добавлено с символом "_" или "." к имени программы (опция). Пример: "WELLE" "WELLE_SPF" "WELLE.SPF"

Указание пути: Краткие обозначения

При указании пути можно использовать следующие условные обозначения:

- CF-карта пользователя: "CF_CARD:"
- USB-диск (панель оператора): "USB:"

Пример

Главная программа "MAIN.MPF" находится в памяти ЧПУ и выбрана для выполнения.

Подпрограмма "SP_1"

Внешняя подпрограмма "SP_1.SPF" или "SP_1.MPF" находится на CF-карте пользователя в директории "/WKS.DIR/WST1.WPD".

Путь к внешней директории программ устанавливается с:

SD42700 \$SC_EXT_PROG_PATH = CF_CARD:WKS.DIR/WST1.WPD

Примечание

Указание пути для вызова внешней подпрограммы:

- Без использования предустановки: "CF_CARD:WKS.DIR/WST1.WPD/SP_1"
- С использованием предустановки: "SP_1"

Подпрограмма "SP_2"

Внешняя подпрограмма "SP_2.SPF" или "SP_2.MPF" находится в директории WKS.DIR/WST1.WPD диска USB. Предустановка пути к внешней директории программ используется для пути подпрограммы "SP_1" и также не переписывается в главной программе. Поэтому при вызове подпрограммы "SP_2" должен быть указан полный путь.

Главная программа "MAIN"**Программный код**

```

N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30

```

Дополнительная информация**Вызов EXTCALL с абсолютным указанием пути**

Если подпрограмма существует по указанному пути, то она выполняется с вызовом EXTCALL. Если подпрограмма отсутствует по указанному пути, то выполнение программы отменяется с вызовом EXTCALL.

Вызов EXTCALL с относительным указанием пути / без указания пути

При вызове EXTCALL с относительным указанием пути или без указания пути поиск в имеющейся программной памяти осуществляется по следующему образцу:

1. Если в SD42700 \$SC_EXT_PROG_PATH предустановлено указание пути, то сначала, исходя из этого пути, выполняется поиск данных в вызове EXTCALL (имя программы возможно с относительным указанием пути). В этом случае абсолютный путь получается через соединение символов:
 - Предустановленное указание пути в SD42700 \$SC_EXT_PROG_PATH
 - Разделительный символ "/"
 - Указание пути и имя подпрограммы в команде EXTCALL
2. Если подпрограмма не была найдена в 1, то поиск выполняется в директориях памяти пользователя.

Поиск прекращается при первом нахождении подпрограммы. Если подпрограмма не найдена, то выполнение программы отменяется с вызовом EXTCALL.

Настраиваемая память догрузки (буфер FIFO)

Для выполнения внешней подпрограммы потребуется память догрузки. Размер памяти догрузки предустановлен (см. MD18360 MM_EXT_PROG_BUFFER_SIZE).

Примечание**Подпрограммы с операторами переходов**

Для внешних подпрограмм, содержащих операторы перехода (GOTO, GOTOB, CASE, FOR, LOOP, WHILE, REPEAT, IF, ELSE, ENDIF и т. п.), цели перехода должны находиться в памяти догрузки.

Примечание**Программы ShopMill-/ShopTurn**

Программы ShopMill и ShopTurn из-за прикрепленных на конце файла описаний контура должны целиком лежать в памяти догрузки.

Для выполняемых параллельно внешних подпрограмм требуется своя память догрузки.

Reset / конец программы / POWER ON

RESET и POWER ON отменяют внешние вызовы подпрограмм и соответствующая память догрузки стирается.

Выбранная для "Выполнения с внешнего устройства" подпрограмма остается выбранной для "Выполнения с внешнего устройства" и после Reset, завершения программы или POWER ON. Поведение не отличается от внутренне выбранных программ, если внешняя программная память остается доступной.

Литература

Дополнительную информацию по "Выполнению с внешнего устройства" можно найти в:

Описание функций "Основные функции"; ГРР, канал, программный режим, параметры Reset (K1)

2.25 Техника макросов (DEFINE ... AS)

ВНИМАНИЕ

Техника макросов повышает комплексность программирования.

С помощью техники макросов язык программирования СЧПУ может быть сильно изменен. При использовании техники макросов необходимо соблюдать осторожность.

Макросом обозначается объединение отдельных операторов в новый общий оператор с собственным именем. При вызове макроса при выполнении программы запрограммированные под макроименем операторы выполняются последовательно.

Согласно области действия, т. е. области, в которой действует макроопределение, существуют следующие категории макросов:

- **Локальные макросы**
Локальные макросы — это макросы, которые определены в начале программы ЧПУ, которая на момент выполнения не является главной программой. Они создаются при вызове программы ЧПУ и удаляются при сбросе (Reset) конца программы или следующем запуске СЧПУ. Обращение к локальным макросам возможно только внутри программы ЧПУ, в которой они определены.
- **Глобальные программные макросы**
Глобальные программные макросы — это макросы, которые определены в начале программы ЧПУ, используемой как главная программа. Они создаются при вызове программы ЧПУ и удаляются при сбросе (Reset) конца программы или следующем запуске СЧПУ. Обращение к глобальным программным макросам возможно в главной программе и во всех подпрограммах.

Примечание

Доступность глобальных программных макросов

Определенные в главной программе глобальные программные макросы доступны в подпрограммах только при следующих установленных машинных данных:

MD11120 \$MN_LUD_EXTENDED_SCOPE = 1

При MD11120 = 0 определенные в главной программе глобальные программные макросы доступны только в главной программе.

- **Глобальные макросы**
Глобальные макросы — это спец. для ЧПУ или канала глобальные макросы, определенные в файле определений (макрофайл) и сохраняющиеся после сброса (Reset) конца программы или следующего запуска СЧПУ. Глобальные макросы могут быть вызваны и выполнены в любой главной программе или подпрограмме.

Примечание

Для использования макросов **внешнего** макрофайла в программе ЧПУ необходимо загрузить макрофайл в ЧПУ.

Перед использованием макросы должны быть определены. Соблюдать следующие правила:

- В макросе могут быть определены любые идентификаторы, функции G, M, N и имена подпрограмм L.
- Макроопределение может происходить в начале программы или в отдельном файле определений (макрофайл).
- Локальные глобальные программные макросы определены в начале программы.
- Глобальные макросы должны быть определены в макрофайле, например, _N_DEF_DIR/_N_UMAC_DEF.
- Макросы G-команд могут быть определены как глобальные макросы.
- Функции N и L могут программироваться 2-позиционно.
- Команды M и G могут программироваться 3-позиционно.

Примечание

Запрещено перезаписывать кодовые слова и зарезервированные имена с помощью макросов.

Синтаксис

Определение макроса:

```
DEFINE <Macro_Name> AS <Operation_1> <Operation_2> ...
```

Вызов в программе ЧПУ:

<Макрос_имя>

Объяснение

DEFINE ... AS:	Комбинация кодовых слов для определения макроса
<Макрос_имя>:	Имя макроса В качестве макроимен разрешены только идентификаторы. С помощью макроимени макрос вызывается из программы ЧПУ.
<Операция_1>:	Первая операция программирования в макросе
<Операция_2>:	Вторая операция программирования в макросе

Граничные условия

Вложенность

Вложенность макросов невозможна.

Примеры

Пример 1: Макроопределение в начале программы

Программный код	Комментарий
DEFINE LINIE AS G1 G94 F300	; макроопределение
...	
N70 LINIE X10 Y20	; макровывоз
...	

Пример 2: Макроопределения в макрофайле

Программный код	Комментарий
DEFINE M6 AS L6	; При смене инструмента вызывается подпрограмма, берущая на себе необходимую передачу данных. В подпрограмме выводится сама функция M смены инструмента (например, M106).
DEFINE G81 AS DRILL(81)	; Эмуляция DIN-G-команды.
DEFINE G33 AS M333 G333	; При резьбонарезании запрашивается синхронизация с PLC. Первоначальная G-команда 33 была через MD переименована в G333, для пользователя программирование остается прежним.

Пример 3: Внешний макрофайл

После загрузки внешнего макрофайла в СЧПУ, макрофайл должен быть загружен в ЧПУ. Только после этого можно использовать макросы в программе ЧПУ.

Программный код	Комментарий
%_N_UMAC_DEF	
;\$PATH=/_N_DEF_DIR	; Спец. для заказчика макросы
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; Правое вращение шпинделя, включение СОЖ
DEFINE M14 AS M4 M7	; Левое вращение шпинделя, включение СОЖ
DEFINE M15 AS M5 M9	; Останов шпинделя, выключение СОЖ
DEFINE M6 AS L6	; Вызов программы смены инструмента
DEFINE G80 AS MCALL	; Отмена цикла сверления
M30	

Управление файлами и программами

3.1 Программная память

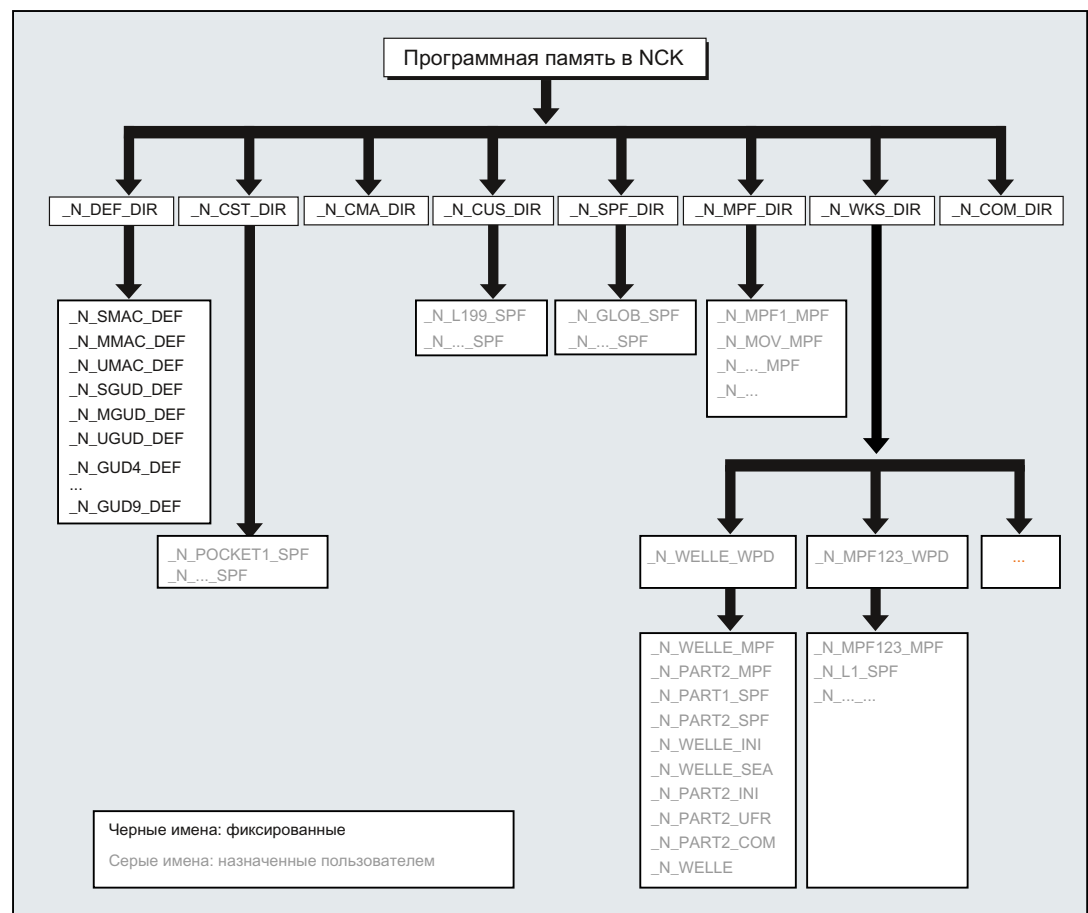
3.1.1 Программная память в ЧПУ

В программной памяти файлы и программы (к примеру, главные и подпрограммы, макроопределения) сохраняются постоянно (→ пассивная файловая система).

Литература

Описание функций "Расширенные функции"; Конфигурация памяти (S7)

Наряду с этим существует часть типов файлов, которые могут быть здесь буферизированы и при необходимости (к примеру, при обработке определенной детали) передаваться в оперативной памяти (к примеру, для инициализации).



Стандартные директории

Следующие директории имеются стандартно:

Директория	Содержание
_N_DEF_DIR	Блоки данных и макроблоки
_N_CST_DIR	Стандартные циклы
_N_CMA_DIR	Циклы изготовителя
_N_CUS_DIR	Циклы пользователя
_N_WKS_DIR	Детали
_N_SPF_DIR	Глобальные подпрограммы
_N_MPF_DIR	Главные программы
_N_COM_DIR	Комментарии

Типы файлов

В программной памяти могут быть размещены следующие типы файлов:

Тип файла	Описание
<имя>_MPF	Главная программа
<имя>_SPF	Подпрограмма
<имя>_TEA	Машинные данные
<имя>_SEA	Установочные данные
<имя>_TOA	Коррекции инструмента
<имя>_UFR	Смещения нулевой точки/фрейм
<имя>_INI	Файл инициализации
<имя>_GUD	Глобальные данные пользователя
<имя>_RPA	R-параметры
<имя>_COM	Комментарий
<имя>_DEF	Определения для глобальных данных пользователя и макросов

Главная директория деталей (_N_WKS_DIR)

Главная директория деталей по умолчанию называется _N_WKS_DIR и находится в программной памяти. Главная директория детали содержит для всех запрограммированных деталей соответствующие каталоги деталей.

Директории деталей (..._WPD)

Директория детали содержит все файлы, необходимые для обработки детали. Это могут быть главные программы, подпрограммы, любые программы инициализации и файлы комментариев.

Программы инициализации выполняются один раз после выбора программы при первом запуске программы обработки детали (согласно машинным данным MD11280 \$MN_WPD_INI_MODE).

Пример:

Директория детали _N_WELLE_WPD, созданная для детали WELLE, содержит следующие файлы:

Файл	Описание
_N_WELLE_MPF	Главная программа
_N_PART2_MPF	Главная программа
_N_PART1_SPF	Подпрограмма
_N_PART2_SPF	Подпрограмма
_N_WELLE_INI	Общая программа инициализации данных для детали
_N_WELLE_SEA	Программа инициализации установочных данных
_N_PART2_INI	Общая программа инициализации данных для программы Part 2
_N_PART2_UFR	Программа инициализации для фрейм-данных для программы Part 2
_N_WELLE_COM	Файл комментария

Дополнительно в директории детали можно хранить и такие данные, которые не требуются ЧПУ для непосредственной обработки. Это могут быть не только файлы ASCII, но и двоичные файлы, например изображения в формате JPG или описания в формате PDF. Чтобы ЧПУ могло интерпретировать их как двоичные файлы, расширения этих файлов должны быть известны в ЧПУ (настройка при вводе в эксплуатацию через MD17000 \$MN_EXTENSIONS_OF_BIN_FILES; по умолчанию предустановлены следующие расширения файлов: JPG, GIF, PNG, BMP, PDF, ICO, HTM).

Выбрать деталь для обработки

Директория детали может быть выбрана для выполнения в одном канале. Если в этой директории находится главная программа **с тем же именем** или только единственная главная программа (_MPF), то она автоматически выбирается для обработки.

Литература:

Руководство оператора

3.1.2 Внешняя программная память

Наряду с пассивной файловой системой в ЧПУ на станке может быть доступна и внешняя программная память (например, на локальном или сетевом диске).

С помощью функций "Выполнение с внешнего устройства" или "EES (Execution from External Storage)" возможно выполнение программ обработки детали **напрямую** из внешней программной памяти.

Литература:

Описание функций "Основные функции"; K1: GPP, канал, программный режим, реакция на Reset

Глобальная память программ обработки деталей (GDIR)

При описании дисков один из них можно отметить как глобальную память программ обработки деталей (GDIR).

Литература:

Руководство оператора; глава: "Работа с программами" > "Настройка дисков"

В этом случае система автоматически создает на диске директории MPF.DIR, SPF.DIR и WKS.DIR. Эти три директории образуют память GDIR.

Память GDIR играет роль исключительно для функции EES. В зависимости от конфигурации диска память GDIR заменяет или расширяет память программ обработки деталей ЧПУ. Тем не менее, настройка памяти GDIR для режима EES не обязательна.

Директории и файлы памяти GDIR могут адресоваться в программе обработки детали точно так же, как в пассивной файловой системе. При этом возможен совместимый перенос программы ЧПУ с указанием пути из пассивной файловой системы в память GDIR. Директория SPF.DIR памяти GDIR содержится в маршруте поиска для подпрограмм.

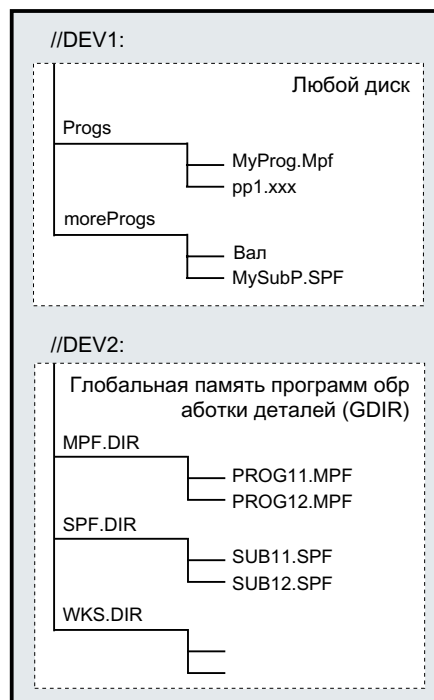
Организация программы

На следующем рисунке показана организация программы во внешней программной памяти:

Пассивная файловая система в NCK



Внешняя программная память



Не зависящие от регистра файловые системы

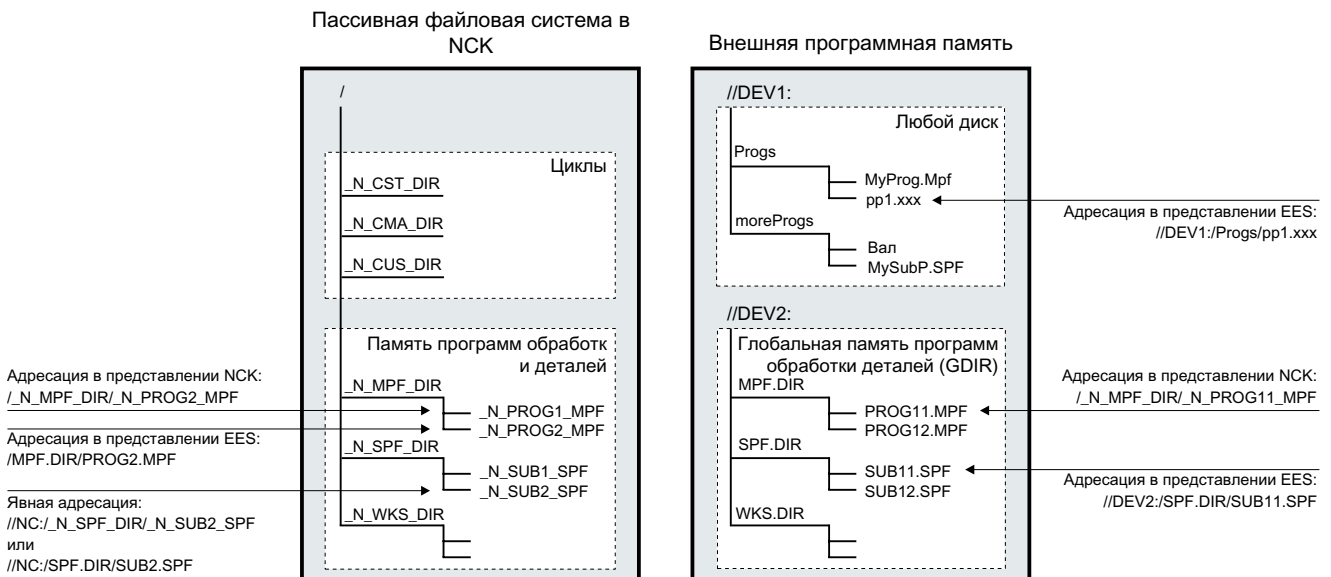
Примечание

Чтобы избежать проблем с прописным и строчным написанием при адресации файлов (см. "Адресация файлов программной памяти (Страница 227)", в качестве внешней программной памяти следует использовать файловые системы, **не зависящие от регистра**.

3.1.3 Адресация файлов программной памяти

На файл в программной памяти, который запрашивается через команду для работы с файлами (например WRITE, DELETE, READ, ISFILE, FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO), дается отсылка либо через абсолютный путь плюс имя файла, либо только через имя файла. Во втором случае в качестве пути файла используется путь выбранной программы.

Адресация в представлении ЧПУ/EES



Адресация файлов пассивной файловой системы

Адресация файлов пассивной файловой системы обычно осуществляется в **представлении ЧПУ** (имена директорий и файлов начинаются с идентификатора домена "_N_", разделительным символом для расширения директории/файла является "_") без указания имени диска. Но и адресация в **представлении EES** (без идентификатора домена "_N_", разделительным символом для расширения директории/файла является ".") тоже допустима.

3.1 Программная память

Пример:

- Представление ЧПУ: `"/_N_SPF_DIR/_N_SUB1_SPF"`
- Представление EES: `"/SPF.DIR/SUB1.SPF"`

Примечание

Адресации файлов пассивной файловой системы в представлении EES преобразуются внутри в представление ЧПУ по следующим правилам:

- К именам директорий и файлов добавляется идентификатор домена `"_N_"`.
- Если четвертым с конца символом в имени директории или файла является точка `(".")`, она преобразуется в символ подчеркивания `("_")`.

Через предопределенное имя диска `"//NC:"` в пассивной файловой системе тоже возможна целенаправленная адресация.

Пример:

- Представление ЧПУ: `"//NC:/_N_SPF_DIR/_N_SUB1_SPF"`
- Представление EES: `"//NC:/SPF.DIR/SUB1.SPF"`

Адресация файлов внешней программной памяти

Адресация файлов внешней программной памяти, не отмеченной как память GDIR, должна осуществляться в представлении EES. В начале пути адресации необходимо указать имя диска (например, `"//DEV1:"`). Допустимы все символические имена устройств, сконфигурированные в `/user/sinumerik/hmi/cfg/logdrive.ini`.

Пример:

- Представление EES: `"//DEV1:/MyProgDir/pp1.xxx"`
- Представление ЧПУ: недопустимо

Адресация файлов глобальной памяти программ обработки деталей (GDIR)

При адресации файлов памяти GDIR наряду с указанием пути в представлении EES допускается и указание пути в представлении ЧПУ.

Пример:

- Представление EES: `"//DEV2:/MPF.DIR/PROG11.MPF"`
- Представление ЧПУ: `"/_N_MPF_DIR/_N_PROG11_MPF"`

Примечание

Адресации файлов памяти GDIR в представлении ЧПУ преобразуются внутри в представление EES по следующим правилам:

- Идентификатор домена `"_N_"` в имени директории и файла удаляется.
- Если четвертым с конца символом в имени директории или файла является символ подчеркивания `("_")`, он преобразуется в точку `(".")`.

Правила указания пути

Полное указание пути состоит из имени диска, пути директории и имени файла.

Имя диска

Для указания имени диска действуют следующие правила:

- Допустимы все символические имена устройств, сконфигурированные в /user/sinumerik/hmi/cfg/logdrive.ini.
- В начале стоит символ "//", за ним следует как минимум одна буква или цифра.
- Последующими символами могут быть буквы, цифры, "_" и пробел в любой комбинации.
- Завершается имя буквой или цифрой, за которой следует двоеточие ":".
- Прочие специальные символы запрещены.

Примечание

Для пассивной файловой системы предопределено имя диска "//NC:".

Примеры:

- Внешняя программная память:
 - //Drive1:
 - //Drive_1:
 - //Drive 1:
 - //A B:
 - //1 B C 2:

Путь директории

Для указания пути директории действуют следующие правила:

- В начале и в конце пути директории и в качестве разделительного символа для отдельных участков пути используется "/".

Примечание

Двойная косая черта ("/") внутри пути директории **недопустима!**

- Имена директорий:
 - Имена директорий должны начинаться с буквы или цифры. Только при адресации в представлении ЧПУ имена директорий начинаются с идентификатора домена "_N_".
 - Последующими символами могут быть буквы, цифры и "_" в любой комбинации.

Примечание

В случае внешней программной памяти в имени директории допускаются и пробелы. Но это не действительно, если внешняя программная память настроена как глобальная память программ обработки деталей (GDIR).

- Прочие специальные символы запрещены.
- Расширения директорий:
 - Расширения директорий должны состоять точно из трех букв/цифр.
 - Они отделяются от имени директории символом "." (представление ЧПУ) или "." (представление EES).

Примечание

В пассивной файловой системе есть только расширения директорий _DIR и _WPD.

Примеры:

- Пассивная файловая система или память GDIR:
 - Представление ЧПУ: _N_WKS_DIR/_N_MYNCPROGS_WPD/...
 - Представление EES: WKS.DIR/MYPROGS.WPD/...
- Внешняя программная память:
 - /abc
 - /ab_c.def
 - /ab c1.def
 - /a b c .d11
 - /abc.def/ghi.klm

Имя файла

Для имен файлов действуют следующие правила:

- Только при адресации в представлении ЧПУ имена файлов начинаются с идентификатора домена "_N_".
- Оба следующих символа должны быть две буквы или знак подчеркивания с буквой после него.

Примечание

Если это условие выполнено, программа ЧПУ может быть вызвана из другой программы как подпрограмма путем указания имени программы. Если имя программы начинается с цифр, вызов подпрограммы возможен только через оператор CALL.

-
- Последующими символами могут быть буквы, цифры и "_" в любой комбинации.
 - Расширение файла:
 - Расширение файла должно состоять точно из трех букв/цифр.

Примечание

Допустимые расширения файлов в пассивной файловой системе представлены в "Программная память в ЧПУ (Страница 223)".

-
- Оно отделяется от имени файла символом "_" (представление ЧПУ) или "." (представление EES).

Примеры:

- Пассивная файловая система или память GDIR:
 - Представление ЧПУ: _N_SUB1_SPF
 - Представление EES: SUB1.SPF
- Внешняя программная память:
 - Teil1
 - _Teil1
 - Teil_1.spf
 - Teil1.mpf

Имя подпрограммы в коде DIN

Для имен подпрограмм в коде DIN действуют следующие правила:

- Первый символ – буква "L".
- Следующие символы – цифры (не менее одной).
- Расширение файла:
 - Расширение файла должно состоять точно из трех букв.
 - Оно отделяется от имени файла символом "_" (представление ЧПУ) или "." (представление EES).

3.1 Программная память

Примеры:

- L123
- L1_SPF (представление ЧПУ) или L1.SPF (представление EES)

Максимальная длина пути

Для указания имени диска и пути директории доступно не более 128 байт, длина имени файла должна составлять не более 31 байт. Следовательно, общая длина пути не должна превышать 159 байт.

3.1.4 Маршрут поиска при вызове подпрограммы

При вызовах подпрограмм без указания пути абсолютный путь определяется через выполнение фиксированного маршрута поиска.

При этом поиск в программной памяти осуществляется в следующей последовательности:

	Директория	Описание
1	текущая директория / <i>name</i>	Текущая директория – это директория, в которой осуществлен выбор программы. Это может быть: • директория детали или стандартная директория <i>_N_MPF_DIR</i> в памяти программ обработки деталей ЧПУ или в глобальной памяти программ обработки деталей или • любая директория внешней программной памяти
2	текущая директория / <i>name_SPF</i>	
3	текущая директория / <i>name_MPF</i>	
4	a //NC:/_N_SPF_DIR / <i>name_SPF</i>	Директория подпрограммы в памяти программ обработки деталей ЧПУ
	B //DEV2:/_N_SPF_DIR / <i>name_SPF</i> ¹⁾	Директория подпрограммы в глобальной памяти программ обработки деталей Указание: Этот этап поиска отсутствует, если глобальная память программ обработки деталей не создана или выбор программы осуществлен в памяти программ обработки деталей ЧПУ.
5	Расширение маршрута поиска, запрограммированное с <i>CALLPATH</i> (см. "Расширить маршрут поиска для вызовов подпрограмм (<i>CALLPATH</i>)" (Страница 210)). Примечание: Этот этап поиска отсутствует, если команда <i>CALLPATH</i> не запрограммирована.	
6	/_N_CUS_DIR / <i>name_SPF</i>	Директория циклов пользователя
7	/_N_CMA_DIR / <i>name_SPF</i>	Директория циклов изготовителя
8	/_N_CST_DIR / <i>name_SPF</i>	Директория стандартных циклов

¹⁾ //DEV2:" означает, например, диск, на котором создана глобальная память программ обработки деталей.

Для поиска действуют следующие правила:

- Маршрут поиска проходится для каждого отдельного вызова подпрограммы, т. е. не важно, где находится программа верхнего уровня.
- В зависимости от директории учитываются разные типы файлов.
- Поиск принципиально не выполняется в директориях нижнего уровня, т. е. во вложенных директориях.

3.1.5 Запрос пути и имени файла

Для запроса пути и имени файла программы ЧПУ предусмотрены следующие системные переменные, считываемые в программе обработки детали:

Системная переменная	Тип	Объяснение
\$P_STACK	INT	Выводит программный уровень, на котором обрабатывается текущая программа ЧПУ.
\$P_PATH[<n>]	STRING	Выводит путь программы ЧПУ, которая обрабатывается на программном уровне, выбранном через индекс поля <n>. Примеры: \$P_PATH[0] выводит путь для главной программы, например "/_N_WKS_DIR/_N_WELLE_WPD/". \$P_PATH[\$P_STACK - 1] выводит путь вызывающей программы. Если путь относится к программе ЧПУ, которая заложена в пассивной файловой системе ЧПУ или в глобальной памяти программ обработки деталей (GDIR), то этот путь выводится в представлении ЧПУ. Если путь относится к программе ЧПУ, которая обрабатывается какой-либо другой внешней программной памятью, но не глобальной памятью программ обработки деталей, то \$P_PATH выводит этот путь в представлении EES.
\$P_PROG[<n>]	STRING	Выводит имя программы ЧПУ, которая обрабатывается на программном уровне, выбранном через индекс поля <n>. Если программа ЧПУ заложена в пассивной файловой системе ЧПУ или в глобальной памяти программ обработки деталей, имя программы выводится в представлении ЧПУ. Если программа ЧПУ обрабатывается с какого-либо другого внешнего диска, но не глобальной памятью программ обработки деталей, то \$P_PROG выводит имя в представлении EES.
\$P_PROGPATH	STRING	Выводит путь программы ЧПУ, которая обрабатывается в данный момент. Вызов \$P_PROGPATH идентичен \$P_PATH[\$P_STACK].

3.1 Программная память

Системная переменная	Тип	Объяснение	
\$P_IS_EES_PATH[<n>]	BOOL	Запрос, соответствует ли выведенный через \$P_PATH[<n>] путь или выведенное через \$P_PROG[<n>] имя программы представлению ЧПУ или представлению EES.	
		= FALSE	\$P_PATH[<n>] и \$P_PROG[<n>] обеспечивают представление ЧПУ. Т. е. перед каждым идентификатором стоит префикс "_N_". Разделительным символом для расширения файла является "_". Примеры: - Путь в представлении ЧПУ: "/_N_WKS_DIR/_N_MYWPD_WPD/" - Имя программы в представлении ЧПУ: "_N_MYPROG_MPF" Путь в представлении ЧПУ может относиться как к пассивной файловой системе в ЧПУ, так и к глобальной памяти программ обработки деталей.
		=TRUE	\$P_PATH[<n>] и \$P_PROG[<n>] выводят представление EES. Т. е. перед идентификаторами префикса "_N_" нет. Разделительным символом для расширения файла является ".". Примеры: - Путь в представлении EES: "//DEV1:/WKS.DIR/MYWPD.WPD/" - Имя программы в представлении EES: "MYPROG.MPF"

<n>: Индекс <n> определяет программный уровень, с которого должна считываться информация о пути (диапазон значений: 0 ... 17)

Примечание

В режиме EES вне глобальной памяти программ обработки деталей (GDIR) системными переменными \$P_PROG, \$P_PATH, и \$P_PROGPATH имена пути выводятся в представлении EES. Поэтому программы пользователя, которые используют и далее обрабатывают эти имена пути, нужно расширить для режима EES таким образом, чтобы они могли обрабатывать и имена пути в представлении EES.

3.2 Оперативная память (CHANDATA, COMPLETE, INITIAL)

Функция

Оперативная память содержит текущие системные данные и данные пользователя, с которыми работает СЧПУ (активная файловая система), например:

- Активные машинные данные
- Данные коррекции инструмента
- Смещения нулевой точки
- ...

Программы инициализации

Здесь речь идет о программах, с которыми предустанавливаются (инициализируются) данные оперативной памяти. Для этого могут использоваться следующие типы файлов:

Тип файла	Описание
name_TEA	Машинные данные
name_SEA	Установочные данные
name_TOA	Коррекции инструмента
name_UFR	Смещения нулевой точки/фрейм
name_INI	Файл инициализации
name_GUD	Глобальные данные пользователя
name_RPA	R-параметры

Области данных

Данные могут быть разбиты на различные области, в которых они должны действовать. Например, СЧПУ может иметь несколько каналов или обычно и несколько осей.

Существует:

Идентификатор	Области данных
ЧПУ	Специфические данные ЧПУ
CH<n>	Специфические данные канала (<n> указывает номер канала)
AX<n>	Специфические данные осей (n указывает номер оси станка)
TO	Данные инструмента
COMPLETE	Все данные

Создание программы инициализации на внешнем РС

С помощью идентификатора области данных и идентификатора типа данных можно определить области, которые при сохранении данных рассматриваются как блок:

_N_AX5_TEA_INI	Машинные данные для оси 5
_N_CH2_UFR_INI	Фреймы канала 2
_N_COMPLETE_TEA_INI	Все машинные данные

После ввода в эксплуатацию СЧПУ в оперативной памяти имеется блок данных, обеспечивающий правильную работу СЧПУ.

Принцип действий для многоканальных СЧПУ (CHANDATA)

CHANDATA (номер канала) для нескольких каналов разрешен только в файле _N_INITIAL_INI. Это файл ввода в эксплуатацию, с помощью которого инициализируются все данные СЧПУ.

Программный код	Комментарий
%_N_INITIAL_INI	
CHANDATA (1)	
	; согласование осей станка канал 1:
\$MC_AXCONF_MACHAX_USED[0]=1	
\$MC_AXCONF_MACHAX_USED[1]=2	
\$MC_AXCONF_MACHAX_USED[2]=3	
CHANDATA (2)	
	; согласование осей станка канал 2:
\$MC_AXCONF_MACHAX_USED[0]=4	
\$MC_AXCONF_MACHAX_USED[1]=5	
CHANDATA (1)	
	: осевые машинные данные:
	; окно точного останова грубого:
\$MA_STOP_LIMIT_COARSE[AX1]=0.2	; ось 1
\$MA_STOP_LIMIT_COARSE[AX2]=0.2	; ось 2
	; окно точного останова точного:
\$MA_STOP_LIMIT_FINE[AX1]=0.01	; ось 1
\$MA_STOP_LIMIT_FINE[AX1]=0.01	; ось 2

ВНИМАНИЕ

Оператор CHANDATA

В программе обработки детали оператор CHANDATA может быть установлен только для канала, в котором выполняется программа ЧПУ. Это означает, что оператор может использоваться для защиты программ ЧПУ от выполнения в не предусмотренном для этого канале.

В случае ошибки выполнение программы отменяется.

Примечание

Файлы INI в списках заданий не содержат операторов CHANDATA.

Сохранение программ инициализации (COMPLETE, INITIAL)

Файлы оперативной памяти могут быть сохранены на внешний PC и снова загружены с него.

- Файлы сохраняются с помощью COMPLETE .
- С помощью INITIAL для всех областей создается файл INI (_N_INITIAL_INI).

Загрузка программ инициализации**ВНИМАНИЕ****Потеря данных**

Если загружается файл с именем "INITIAL_INI", то все данные, которые не обеспечиваются в файле, инициализируются со стандартными данными. Исключением являются только машинные данные. Т.е **установочные данные, данные инструмента, WO, значения GUD, ...** обеспечиваются со стандартными данными (обычно "НОЛЬ").

Для загрузки отдельных машинных данных, подходит, например, файл COMPLETE_TEA_INI. В этом файле СЧПУ ожидает только машинные данные. Тем самым другие области данных в этом случае не затрагиваются.

Загрузка программ инициализации

Программы INI могут выбираться и вызываться и как программы обработки детали, если они используют только данные одного канала. Таким образом также возможна инициализация управляемых программой данных.

Защищенные области

4.1 Определение защищенных областей (CPROTDEF, NPROTDEF)

Защищенные области, которые должны защищать элементы станка от столкновений, определены соответственно в блоках в программе обработки детали. Они включают следующие элементы:

1. Определение рабочей плоскости
Перед собственно началом определения защищенной области необходимо выбрать рабочую плоскость, к которой должно относиться описание контура защищенной области.
2. Начало определения
В зависимости от команды ЧПУ защищенная область создается либо спец. для канала, либо спец. для станка.
3. Описание контура защищенной области
Контур защищенной области описывается посредством движений перемещения. Они не выполняются и не связаны с предшествующими или последующими геометрическими описаниями. Они определяют только защищенную область.
4. Конец определения

Синтаксис

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF (<n>, <t>, <AppLim>, <AppPlus>, <AppMinus>)
G0/G1/... X/Y/Z...
...
EXECUTE (<Var>)
```

Объяснение

DEF INT <Var>:	Определение локальной вспомогательной переменной (тип данных: INTEGER)
<Var>:	Имя вспомогательной переменной
G17/G18/G19:	Рабочая плоскость Указание: Рабочая плоскость не может изменяться до конца определения. Программирование аппликаты между началом и концом определения не допускается.
CPROTDEF ():	Предопределенная процедура определения спец. для канала защищенной области
NPROTDEF ():	Предопределенная процедура определения спец. для станка защищенной области

4.1 Определение защищенных областей (CPROTDEF, NPROTDEF)

<n>:	Номер определенной защищенной области		
	Тип данных:	INT	
<t>:	Тип защищенной области		
	Тип данных:	BOOL	
	Значение:	TRUE	Относящаяся к инструменту защищенная область
		FALSE	Относящаяся к детали защищенная область
<AppLim>:	Тип ограничения в 3-м измерении		
	Тип данных:	INT	
	Значение:	0	нет ограничения
		1	ограничение в плюсовом направлении
		2	ограничение в минусовом направлении
		3	Ограничение в плюсовом и минусовом направлении
<AppPlus>:	Значение ограничения в плюсовом направлении 3-й размерности		
	Тип данных:	REAL	
<AppMinus>:	Значение ограничения в минусовом направлении 3-й размерности		
	Тип данных:	REAL	

4.1 Определение защищенных областей (CPROTDEF, NPROTDEF)

G0/G1/... X/Y/Z... ...:	Контур защищенной области описывается макс. 11 движениями перемещения в выбранной рабочей плоскости. При этом первое движение перемещения — это движение к контуру. Последняя точка описания контура всегда должна совпадать с первой точкой описания контура. Защищенной областью считается область слева от контура: • внутренняя защищенная область Контур для внутренней защищенной области следует описывать против часовой стрелки. • Наружная защищенная область (допускается только для относящихся к детали защищенных областей) Контур для наружной защищенной области следует описывать по часовой стрелке. Допустимы следующие элементы контура: • G0, G1 для прямых элементов контура • G2 для сегментов окружности по часовой стрелке Допустимо только для относящихся к детали защищенных областей! Не допустимо для относящихся к инструменту защищенных областей, т. к. они могут быть только выпуклыми. • G3 для круговых сегментов против часовой стрелки Указание: Защищенная область не может быть описана полным кругом. Полный круг должен быть разбит на две делительные окружности. Указание: Последовательность G2 → G3 или G3 → G2 не допустима! Между двумя круговыми кадрами должен быть вставлен короткий кадр G1.
EXECUTE (<Var>):	Предопределенная процедура, отмечающая конец определения EXECUTE возвращает к обычной обработке программы.

Пример

Пример см. в "Активация/деактивация защищенных областей (CPROT, NPROT) (Страница 244)".

Дополнительная информация**Спец. для станка защищенные области**

Относящаяся к станку защищенная область или ее контур определяется посредством гео-осей, т. е. относительно базовой кинематической системы (BKS) канала. Чтобы был возможен правильный контроль защищенных областей во всех каналах, в которых активна спец. для станка защищенная область, базовые кинематические системы (BKS) всех участвующих каналов должны быть идентичны:

- положение начала координат относительно нулевой точки станка
- Ориентация осей координат

Исходная точка описания контура

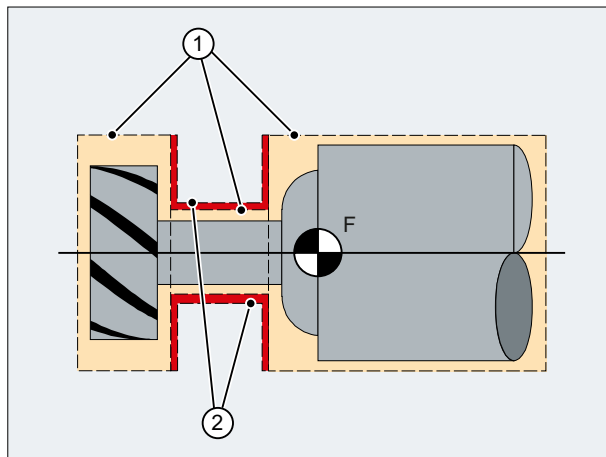
- Относящиеся к инструменту защищенные области
Координаты для относящихся к **инструменту** защищенных областей указываются абсолютно, относительно **опорной точки инструментального суппорта F**.
- Относящиеся к детали защищенные области
Координаты для относящихся к **детали** защищенных областей указываются абсолютно, относительно **нулевой точки базовой кинематической системы (BKS)**.

Вращательно-симметричные защищенные области

Для вращательно-симметричных защищенных областей (например, патрон шпинделя) необходимо описание всего контура, а не только до центра вращения.

Относящиеся к инструменту защищенные области

Относящиеся к инструменту защищенные области всегда должны быть выпуклыми. Если необходима вогнутая защищенная область, то необходимо разбить его на несколько выпуклых защищенных областей.



- ① Выпуклые защищенные области
- ② Вогнутые защищенные области (**недопустимы!**)
- F Исходная точка инструментального суппорта

Граничные условия

При определении защищенной области следующие функции не могут быть активны и запрещены к использованию:

- коррекция радиуса инструмента (коррекция радиуса фрезы, коррекция радиуса резца)
- трансформация
- движение к референтной точке (G74)
- движение к фиксированной точке (G75)
- время ожидания (G4)
- останов предварительной обработки кадра (STOPRE)

4.1 Определение защищенных областей (CPROTDEF, NPROTDEF)

- конец программы (M17, M30)
- M-функции: M0, M1, M2

4.2 Активация/деактивация защищенных областей (CPROT, NPROT)

Предварительно определенные в программе обработки детали защищенные области можно в любой момент активировать либо предварительно активировать для последующей активации через программу электроавтоматики. Активные защищенные области можно деактивировать в любой момент.

При активации либо предварительной активации возможно относительное смещение исходной точки защищенной области.

Примечание

Защищенная область учитывается только после реферирования всех гео-осей канала, в котором она активирована.

Примечание

Контроль защищенных областей

Если ни одна относящаяся к инструменту защищенная область не активна, то траектория инструмента проверяется по отношению к относящимся к детали защищенным областям.

Если ни одна относящаяся к детали защищенная область не активна, контроль защищенных областей не выполняется.

Синтаксис

CPROT (<n>, <Status>, <XMov>, <YMov>, <ZMov>)

NPROT (<n>, <Status>, <XMov>, <YMov>, <ZMov>)

Объяснение

CPROT:	Предопределенная процедура активации спец. для канала защищенной области			
NPROT:	Предопределенная процедура активации спец. для станка защищенной области			
<n>:	Номер защищенной области			
	Тип данных:	INT		
<состояние>:	С помощью этого параметра устанавливается спец. для канала статус активации.			
	Тип данных:	INT		
	Значение:	0	Деактивировать защищенную область	
		1	предварительная активация защищенной области	
		2	Активировать защищенную область	
		3	Предварительно активировать защищенную область с условным остановом	

<XMov>, <YMov>, <ZMov>:	Значения аддитивного смещения в направлении X/Y/Z Смещение может выполняться в 1, 2 или 3 размерностях. Значения смещения относятся к: • нулевой точке станка при относящейся к детали защищенной области • исходной точке инструментального суппорта F для спец. для инструмента защищенной области.	
	Тип данных:	REAL

Пример

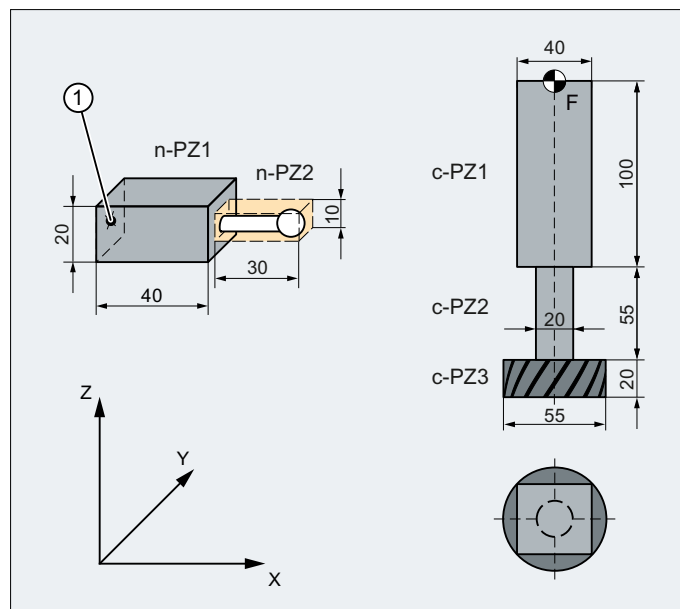
Для фрезерного станка необходимо контролировать возможное столкновение фрезы с измерительным щупом. Положение измерительного щупа при активации должно быть указано через смещение.

Для этого определяются следующие защищенные области:

- Соответственно по одной спец. для станка и относящейся к детали защищенной области для держателя измерительного щупа (n-PZ1) и самого измерительного щупа (n-PZ2).
- Соответственно по одной спец. для станка и относящейся к детали защищенной области для держателя фрезы (с-PZ1), хвостовика фрезы (с-PZ2) и самой фрезы (с-PZ3).

Ориентация всех защищенных областей лежит в направлении Z.

Положение исходной точки измерительного щупа при активации должно находиться на X = -120, Y = 60 и Z = 80.



- ① Исходная точка для защищенной области измерительного щупа
F Исходная точка инструментального суппорта

4.2 Активация/деактивация защищенных областей (CPROT, NPROT)

Программный код	Комментарий
DEF INT PROTZONE	; Определение вспомогательной переменной
G17	; Рабочая плоскость XY
; Определение защищенных областей:	
NPROTDEF(1,FALSE,3,10,-10)	; Защищенная область n-PZ1
G01 X0 Y-10	
X40	
Y10	
X0	
Y-10	
EXECUTE (PROTZONE)	
NPROTDEF(2,FALSE,3,5,-5)	; Защищенная область n-PZ2
G01 X40 Y-5	
X70	
Y5	
X40	
Y-5	
EXECUTE (PROTZONE)	
CPROTDEF(1,TRUE,3,0,-100)	; Защищенная область c-PZ1
G01 X-20 Y-20	
X20	
Y20	
X-20	
Y-20	
EXECUTE (PROTZONE)	
CPROTDEF(2,TRUE,3,-100,-150)	; Защищенная область c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE (PROTZONE)	
CPROTDEF(3,TRUE,3,-150,-170)	; Защищенная область c-PZ3
G01 X0 Y-27.5	
G03 X0 Y27.5 J27.5	
X0 Y27.5 J-27.5	
EXECUTE (PROTZONE)	
; Активация защищенных областей:	
NPROT(1,2,-120,60,80)	; Активация защищенной области n-PZ1 со смещением
NPROT(2,2,-120,60,80)	; Активация защищенной области n-PZ2 со смещением
CPROT(1,2,0,0,0)	; Активация защищенной области c-PZ1
CPROT(2,2,0,0,0)	; Активация защищенной области c-PZ2
CPROT(3,2,0,0,0)	; Активация защищенной области c-PZ3

Дополнительная информация

Статус активации после запуска СЧПУ

Защищенная область может быть активна сразу же после запуска СЧПУ и реферирования осей. Это случай, когда для защищенной области следующая системная переменная установлена на TRUE:

- \$SN_PA_ACTIV_IMMED[<n>] (для спец. для станка защищенной области) или
- \$SC_PA_ACTIV_IMMED[<n>] (для спец. для канала защищенной области)
Индекс "<n>" соответствует номеру защищенной области: 0 = 1 Защищенная область

Защищенная область активируется статусом = 2 и без смещения.

Множественная активация защищенной области

Одна спец. для станка защищенная область может одновременно действовать в нескольких каналах (например, защищенная область пиноли при двух противолежащих салазках). Контроль защищенных областей выполняется только в том случае, если все гео-оси реферированы.

Защищенная область в одном канале не может быть одновременно активирована с различными смещениями.

Контроль защищенной области при активной коррекции радиуса инструмента

При активной коррекции радиуса инструмента работоспособный контроль защищенной области возможен только в том случае, если плоскость коррекции радиуса инструмента идентична плоскости определения защищенной области.

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI)

Функция

Функция CALCPOSI проверяет в системе координат детали (WCS), можно ли перемещать **гео-оси** из стартовой позиции по заданному пути перемещения, не нарушая активных ограничений. Если путь перемещения из-за ограничений не может быть пройден полностью, то возвращается положительное значение состояния в десятичной кодировке и макс. возможный путь перемещения.

Определение

```
INT CALCPOSI (VAR REAL[3] <Start>, VAR REAL[3] <Dist>, VAR REAL[5]
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

Синтаксис

```
<состояние> = CALCPOSI (VAR <Start>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <MeasSys>, <TestLim>)
```

Объяснение

CALCPOSI (...):	Предопределенная функция для теста на нарушение ограничений в отношении гео-осей	
	Остановка предварительной обработки:	нет
	Автономно в кадре:	да

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI)

<состояние>: (часть 1)		Возвращаемое значение функции. Отрицательные значения показывают состояния ошибки.		
		Тип данных: INT		
		Диапазон значений: $-8 \leq x \leq 100000$		
		Значение:	0	Путь перемещения может быть пройден полностью
			-1	В <Limit>, по крайней мере, один компонент — отрицательный.
			-2	Ошибка при описании трансформации. Пример: Путь перемещения проходит через сингулярность, так что позиции осей не определены.
			-3	Указанный путь перемещения <Dist> и максимально возможный путь перемещения <MaxDist> находятся в линейной зависимости. Указание Возможно только в сочетании с <TestLim>, бит 4 == 1.
			-4	Проекция направления перемещения из <Dist> на ограничивающую поверхность — это нулевой вектор, или направление перемещения располагается вертикально на нарушенной ограничивающей поверхности. Указание Возможно только в сочетании с <TestLim>, бит 5 == 1.
			-5	В <TestLim> бит 4 == 1 И бит 5 == 1
			-6	По крайней мере, одна ось станка, которую необходимо рассмотреть для проверки границ перемещения, не реферирована.
			-7	Функция предотвращения столкновений: Недействительное определение кинематической цепочки или защищенных областей.
			-8	Функция предотвращения столкновений: Функция не может быть выполнена из-за недостаточности памяти.

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOS1)

<состояние>: (часть 2)	Первая позиция		
	Указание Если одновременно нарушено несколько границ, то сигнал поступает по той границе, которая вызывает наибольшее ограничение заданного пути перемещения.		
	Значение:	1	Программные конечные выключатели ограничивают путь перемещения
		2	Ограничение рабочего поля ограничивает путь перемещения
		3	Защищенные области ограничивают путь перемещения
		4	Функция предотвращения столкновений: Защищенные области ограничивают путь перемещения
Вторая позиция			
Значение:	1x	начальное значение нарушает границу	
	2x	Заданная прямая нарушает границу. Это значение возвращается и тогда, когда сама конечная точка не нарушает границы, но на пути от стартовой к конечной точке произошло бы нарушение предельного значения (например, прохождение через защищенную область, изогнутые программные конечные выключатели в WCS при нелинейных трансформациях, например, Transmit).	
<состояние>: (часть 3)	Третья позиция		
	Значение:	1xx	И первая позиция == 1 или 2: Положительное предельное значение нарушено.
			И первая позиция == 3 ¹⁾ : Нарушена спец. для ЧПУ защищенная область.
		2xx	И первая позиция == 1 или 2: Отрицательное предельное значение нарушено.
			И первая позиция == 3 ¹⁾ : Нарушена спец. для канала защищенная область.
	Четвертая позиция		
<состояние>: (часть 4)	Значение:	1xxx	И первая позиция == 1 или 2: Коэффициент, на который умножается номер оси, нарушающей границу. Счет осей начинается с 1. Отношение: - Программные конечные выключатели: Оси станка - Ограничение рабочего поля: Геометрические оси И первая позиция == 3 ¹⁾ : Коэффициент, на который умножается номер нарушенной защищенной области.

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI)

<состояние>: (часть 5)	Шестая позиция		
	Значение:	0xxxxx	Шестая позиция == 0: <Dist> не изменяется
		1xxxxx	В <Dist> возвращается вектор направления, определяющий дальнейшее направление движения на ограничивающей поверхности. Возможно только при следующих граничных условиях: - Программные конечные выключатели или ограничение рабочего поля нарушены (не в стартовой точке) - Трансформация не активна <TestID>, бит 4 или бит 5 == 1
<Start>:	Ссылка на вектор со стартовыми позициями: <Start> [0]: 1-й гео-оси <Start> [1]: 2-й гео-оси <Start> [2]: 3-й гео-оси		
	Тип параметра:	Вход	
	Тип данных:	VAR REAL[3]	
	Диапазон значений:	-макс. REAL-значение ≤ x[<n>] ≤ +макс. REAL-значение	
<Dist>:	Ссылка на вектор.		
	Вход: инкрементальная величина пути перемещения <Dist> [0]: 1-й гео-оси <Dist> [1]: 2-й гео-оси <Dist> [2]: 3-й гео-оси		
	Выход (только при установленной шестой позиции в <состояние>):		
		В качестве выходного значения <Dist> содержит единичный вектор v , определяющий дальнейшее направление перемещения в WCS.	
		Случай 1: Формирование вектора v для <TestID>, бит 4 == 1 Входные векторы <Dist> и <MaxDist> образуют плоскость движения. Эта плоскость пересекается с нарушенной ограничивающей поверхностью. Прямая пересечения обеих плоскостей определяет направление вектора v . При этом ориентация (знак) выбирается таким образом, чтобы угол между входным вектором <MaxDist> и v не превышал 90 градусов.	
		Случай 2: Формирование вектора v для <TestID>, бит 5 == 1 Вектор v — единичный вектор в направлении проекции содержащегося в <Dist> вектора перемещения на ограничивающую поверхность. Если проекция вектора перемещения на ограничивающую поверхность представляет собой нулевой вектор, то возвращается ошибка.	
	Тип параметра:	Вход/выход	
	Тип данных:	VAR REAL[3]	
Диапазон значений:	-макс. REAL-значение ≤ x[<n>] ≤ +макс. REAL-значение		

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI)

<Limit>:	Ссылка на поле длиной 5. • <Limit> [0 - 2]: Минимальные отступы гео-осей от ограничений: – <Limit> [0]: 1-й гео-оси – <Limit> [1]: 2-й гео-оси – <Limit> [2]: 3-й гео-оси Все минимальные отступы соблюдены при: – Ограничение рабочего поля: без ограничений – Программные конечные выключатели: Если нет активной трансформации или если активна трансформация, при которой возможно однозначное соотнесение гео-осей с линейными осями станка, например, 5-осевые трансформации. • <Limit> [3]: Содержит минимальный отступ линейных осей станка, согласование которых с гео-осью невозможно, например, из-за нелинейной трансформации. Это значение используется также как предельное значение при контроле обычных защищенных областей и защищенных областей предотвращения столкновений. • <Limit> [4]: Содержит минимальный отступ круговых осей станка, согласование которых с гео-осью невозможно, например, из-за нелинейной трансформации. Указание Это значение действует только при контроле программных конечных выключателей специальных трансформаций.	
	Тип параметра:	Вход
	Тип данных:	VAR REAL[5]
	Диапазон значений:	-макс. REAL-значение ≤ x[n] ≤ макс. REAL-значение
<MaxDist>:	Ссылка на вектор с инкрементальным значением пути перемещения, при котором заданное минимальное расстояние границы оси не нарушает ни одна из задействованных осей: • <Dist> [0]: 1-й гео-оси • <Dist> [1]: 2-й гео-оси • <Dist> [2]: 3-й гео-оси Если путь перемещения не ограничен, то содержание этого возвращаемого параметра идентично содержанию <Dist>. Для <TestID>, бит 4 == 1: <Dist> и <MaxDist> В качестве входных значений <MaxDist> и <Dist> должны содержать векторы, образующие плоскость движения. Оба вектора не должны иметь линейной зависимости друг от друга. Величина <MaxDist> произвольная. Для расчета направления движения см. описание <Dist>.	
	Тип параметра:	Выход
	Тип данных:	VAR REAL[3]
	Диапазон значений:	-макс. REAL-значение ≤ x[<n>] ≤ +макс. REAL-значение

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI)

<MeasSys>:	Система единиц (дюймовая / метрическая) для позиционных и линейных данных (опция)	
	Тип данных:	BOOL
	Значение:	FALSE (по умолчанию) Система единиц согласно текущей активной G-команде из G-группы 13 (G70, G71, G700, G710). Указание При активной G70 и метрической основной системе или активной G71 и дюймовой основной системе системные переменные \$AA_IW и \$AA_MW выводятся в основной системе и при использовании для CALCPOSI при необходимости должны быть пересчитаны.
	TRUE	Система единиц согласно установленной основной системе: MD52806 \$MN_ISO_SCALING_SYSTEM
<TestLim>:	Биткодированный выбор контролируемых ограничений (опция)	
	Тип данных:	INT
	Значение по умолчанию:	Бит 0, 1, 2, 3, 6, 7 == 1 (207)
	Бит	Дес. Объяснение
	0	1 Программные конечные выключатели
	1	2 Ограничение рабочего поля
	2	4 Активированные обычные защищенные области
	3	8 Преактивированные обычные защищенные области
	4	16 При нарушении программных конечных выключателей или ограничений рабочего поля в <Dist> возвращать направление перемещения согласно случаю 1 (см. выше).
	5	32 При нарушении программных конечных выключателей или ограничений рабочего поля в <Dist> возвращать направление перемещения согласно случаю 2 (см. выше).
	6	64 Активированные защищенные области недопущения столкновений
	7	128 Преактивированные защищенные области предотвращения столкновений
	8	256 Пары активированных и преактивированных защищенных областей предотвращения столкновений
¹⁾ Если нарушено несколько защищенных областей, то сигнализируется та, которая вызывает наибольшее ограничение заданного пути перемещения.		

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI)

Программный код

; Относящаяся к инструменту защищенная область C2

N70 CPROTDEF(2, TRUE, 0)

N80 G17 G1 X-2 Y0

N90 G3 I2 X2

N100 I-2 X-2

N110 EXECUTE(_PA)

; Относящаяся к детали защищенная область C4

N120 CPROTDEF(4, FALSE, 0)

N130 G17 G1 X0 Y15

N140 X10

N150 Y25

N160 X0

N170 Y15

N180 EXECUTE(_PA)

; Спец. для станка защищенная область N3

N190 NPROTDEF(3, FALSE, 0)

N200 G17 G1 X10 Y5

N210 X25

N220 Y15

N230 X10

N240 Y5

N250 EXECUTE(_PA)

; Активировать или предварительно активировать защищенные области

N260 CPROT(2, 2, 0, 0, 0)

N270 CPROT(4, 1, 0, 0, 0)

N280 NPROT(3, 2, 0, 0, 0)

; Определить ограничения рабочего поля

N290 G25 XX=-10 YY=-10

N300 G26 XX=20 YY=21

N310 _START[0] = 0.

N320 _START[1] = 0.

N330 _START[2] = 0.

N340 _DIST[0] = 35.

N350 _DIST[1] = 20.

N360 _DIST[2] = 0.

N370 _LIMIT[0] = 0.

N380 _LIMIT[1] = 0.

N390 _LIMIT[2] = 0.

N400 _LIMIT[3] = 0.

N410 _LIMIT[4] = 0.

N420 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST)

N430 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,3)

N440 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,1)

N450 _START[0] = 5.

N460 _START[1] = 17.

N470 _START[2] = 0.

N480 _DIST[0] = 0.

N490 _DIST[1] = -27.

N500 _DIST[2] = 0.

N510 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,14)

N520 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,6)

N530 _LIMIT[1] = 2.

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI)

Программный код

```

N540 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N550 _START[0] = 27.
N560 _START[1] = 17.1
N570 _START[2] = 0.
N580 _DIST[0] = -27.
N590 _DIST[1] = 0.
N600 _DIST[2] = 0.
N610 _LIMIT[3] = 2.
N620 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,12)
N630 _START[0] = 0.
N640 _START[1] = 0.
N650 _START[2] = 0.
N660 _DIST[0] = 0.
N670 _DIST[1] = 30.
N680 _DIST[2] = 0.
N690 TRANS X10
N700 AROT Z45
N710 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST)
; Снова удалить фреймы из N690 и N700
N720 TRANS
N730 _START[0] = 0.
N740 _START[1] = 10.
N750 _START[2] = 0.
; Векторы _DIST и _MAXDIST определяют плоскость движения
N760 _DIST[0] = 30.
N770 _DIST[1] = 30.
N780 _DIST[2] = 0.
N790 _MAXDIST[0] = 1.
N800 _MAXDIST[1] = 0.
N810 _MAXDIST[2] = 1.
N820 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,17)
N830 M30

```

Результаты CALCPOSI()

N...	<состоя- ние>	<MaxDist>[0] Δ X	<MaxDist>[1] Δ Y	Примечания
420	3123	8.040	4.594	N3 нарушается.
430	1122	20.000	11.429	Контроль защищенных областей отсутствует, ограничение рабочего поля нарушается.
440	1121	30.000	17.143	Активен только контроль программных конечных выключателей.
510	4213	0.000	0.000	Нарушение стартовой точки C4
520	0000	0.000	-27.000	Предварительно активированная C4 не контролируется. Заданный путь может быть пройден полностью.
540	2222	0.000	-25.000	Из-за _LIMIT[1] = 2 путь перемещения сокращается ограничением рабочего по- ля.

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI)

N...	<состоя- ние>	<MaxDist>[0] Δ X	<MaxDist>[1] Δ Y	Примечания
620	4223	-13.000	0.000	Интервал до C4 из-за C2 и _LIMIT[3] в общей сложности 4 мм. Интервал C2 → N3 в 0,1 мм не вызывает ограничения пути перемещения.
710	1221	0.000	21.213	Фрейм со смещением и вращением активен. Допустимый путь перемещения в _DIST действителен в смещенной и повернутой WCS.
820	102121	18.000	18.000	Программный конечный выключатель оси Y нарушается. При <_TESTLIM> = 17 запрашивается вычисление направления дальнейшего движения. Это направление указано в <_DIST> (0.707, 0.0, 0.707). Оно действительно, поскольку в <_STATUS> установлена шестая позиция.

Дополнительная информация

Состояние оси "Реферирована"

Все рассматриваемые CALCPOSI () оси станка должны быть реферированы.

Характеристики перемещения относительно окружности

Характеристики перемещения относительно окружности **всегда** интерпретируются как характеристики радиуса. Это следует учитывать, в частности, для поперечных осей с активированным программированием диаметра (DIAMON / DIAM90).

Сокращение пути перемещения

Если указанный путь перемещения оси ограничивается, то в возвращаемом значении <MaxDist> пропорционально сокращаются и пути перемещения других осей. Тем самым результирующая конечная точка и далее расположена на заданной траектории.

Круговые оси

Круговые оси контролируются только в том случае, если они не являются осями по модулю.

Для одной или нескольких участвующих осей допустимо отсутствие определений программных конечных выключателей, ограничений рабочего поля или защищенных областей.

Состояние программного конечного выключателя и ограничения рабочего поля

Программные конечные выключатели и ограничения рабочего поля учитываются, только если они при выполнении CALCPOSI () активны. На состояние можно повлиять, например, через:

- Машинные данные: MD21020 \$MC_WORKAREA_WITH_TOOL_RADIUS
- Установочные данные: \$AC_WORKAREA_CS_...

4.3 Проверка нарушения защищенной области, ограничения рабочего поля и программных конечных выключателей (CALCPOSI)

- Интерфейсные сигналы ЧПУ/PLC: DB31, ... DBX12.2 / 3
- Команды: WALIMON / WALIMOF

Программные конечные выключатели и трансформации

В случае CALCPOSI () при различных кинематических трансформациях (например, TRANSMIT) на основе многозначности в определенных местах диапазона перемещения позиции осей станка (MCS) не всегда можно однозначно определить по позициям геометрических осей (WCS). В обычном режиме перемещения однозначность следует, как правило, из предыстории и условия, что непрерывному движению в WCS должно соответствовать непрерывное движение осей станка в MCS. Поэтому в подобных случаях при контроле программных конечных выключателей для разрешения многозначности используется текущая позиция станка на момент выполнения функции CALCPOSI ().

Примечание

Остановка предварительной обработки

При использовании CALCPOSI () в сочетании с трансформацией только пользователь несет ответственность за программирование для синхронизации позиций станка с предварительной обработкой перед CALCPOSI () остановки предварительной обработки (STOPRE).

Отступ защищенных областей и обычные защищенные области

В случае обычных защищенных областей не гарантируется, что при движении перемещения на заданном пути перемещения указанный в параметре <Limit>[3] безопасный отступ соблюдается в отношении всех защищенных областей. Гарантируется лишь, что при удлинении возвращаемой в <Dist> конечной точки на безопасный отступ в направлении перемещения ни одна защищенная область не нарушается. Но ход прямой может пролегать на любом расстоянии от защищенной области.

Отступ защищенных областей и защищенные области предотвращения столкновений

В случае защищенных областей предотвращения столкновения гарантируется, что при движении перемещения на заданном пути перемещения указанный в параметре <Limit>[3] безопасный отступ соблюдается в отношении всех защищенных областей.

Указанный в параметре <Limit>[3] безопасный отступ действует, только если:

<Limit>[3] > (MD10619 \$MN_COLLISION_TOLERANCE)

Если в параметре <TestLim> установлен бит 4 (вычисление направления дальнейшего перемещения), то содержащийся в <DIST> вектор направления действителен, только если установлена шестая позиция в возвращаемом значении функции (<состояние>). Если такое направление определить невозможно, поскольку либо были нарушены защищенные области, либо активна какая-либо трансформация, входное значение в <DIST> остается неизменным. Дополнительное сообщение об ошибке не выводится.

Специальные команды перемещения

5.1 Подвод к кодированным позициям (CAC, CIC, CDC, CACP, CACN)

С помощью следующих команд через номера позиции можно перемещать линейные и круговые оси на сохраненные в таблицах машинных данных фиксированные позиции осей. Такой тип программирования обозначается как "Подвод к кодированным позициям".

Синтаксис

CAC (<n>)
CIC (<n>)
CACP (<n>)
CACN (<n>)

Значение

CAC (<n>):	Выполнить подвод к кодированной позиции с номером позиции n
CIC (<n>):	Выполнить подвод к кодированной позиции, исходя из актуального номера позиции, на n-количество позиций вперед (+) или назад (–)
CDC (<n>):	Выполнить подвод к кодированной позиции с номером позиции n по кратчайшему пути (только для круговых осей)
CACP (<n>):	Выполнить подвод к кодированной позиции с номером позиции n в положительном направлении (только для круговых осей)
CACN (<n>):	Выполнить подвод к кодированной позиции с номером позиции n в отрицательном направлении (только для круговых осей)
<n>:	Номер позиции в таблице машинных данных Диапазон значений: 0, 1, ... (макс. число мест в таблице - 1)

Пример: Подвод к кодированным позициям позиционирующей оси

Программный код	Комментарий
N10 FA[B]=300	; Подача для позиционирующей оси В
N20 POS[B]=CAC(10)	; Выполнить подвод к кодированной позиции с номером позиции 10
N30 POS[B]=CIC(-4)	; Выполнить подвод к кодированной позиции с "актуальным номером позиции" - 4

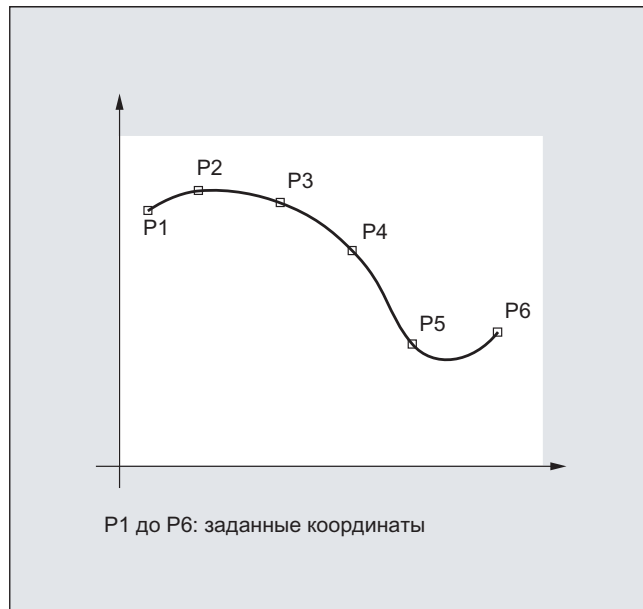
Литература

- Описание функций "Расширенные функции"; делительные оси (T1)
- Описание функций "Синхронные действия"

5.2 Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Аналитически точное описание произвольно изогнутых контуров на деталях невозможно. Поэтому аппроксимация таких контуров выполняется через ограниченное число опорных точек, к примеру, при оцифровке поверхностей. Для создания оцифрованной поверхности на детали опорные точки должны быть объединены в одно описание контура. Это обеспечивает сплайн-интерполяция.

Сплайн определяет кривую, состоящую из полиномов 2-го или 3-го порядка. Свойства на опорных точках сплайна могут определяться **в зависимости от используемого типа сплайна**.



Следующие типы сплайнов доступны для SINUMERIK solution line:

- А-сплайн
- В-сплайн
- С-сплайн

Синтаксис

Общая информация:

```
ASPLINE X... Y... Z... A... B ... C ...
BSPLINE X... Y... Z... A... B ... C ...
CSPLINE X... Y... Z... A... B ... C ...
```

Для В-сплайна дополнительно можно запрограммировать:

PW=<n>

SD=2

PL=<значение>

Для А- и С-сплайна дополнительно можно запрограммировать:

BAUTO / BNAT / BTAN

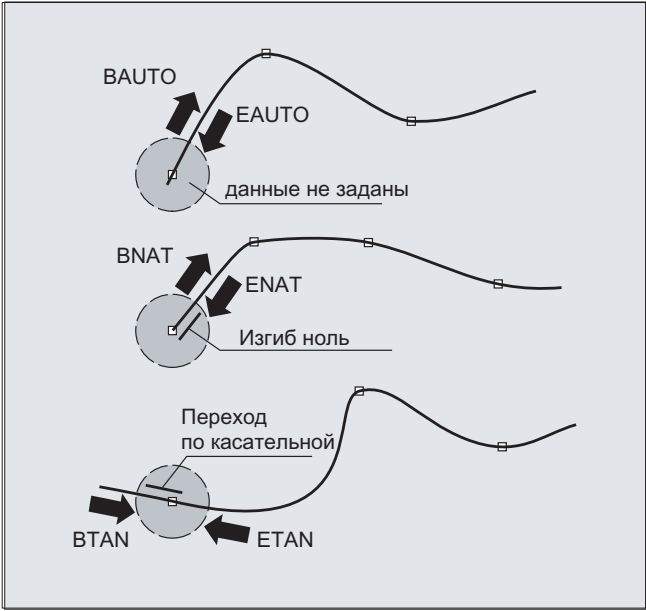
5.2 Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

EAUTO / ENAT / ETAN

Значение

Тип сплайн-интерполяции:				
ASPLINE:		Команда для включения А-сплайн-интерполяции		
BSPLINE:		Команда для включения В-сплайн-интерполяции		
CSPLINE:		Команда для включения С-сплайн-интерполяции		
		Команды ASPLINE, BSPLINE и CSPLINE действуют модально и относятся к группе команд перемещения.		
Опорные точки или контрольные точки:				
X... Y... Z... A... B ... C...		Позиции в декартовых координатах		
Вес точки (только В-сплайн):				
PW:		С помощью команды PW для каждой опорной точки можно запрограммировать т.н. "Вес точки".		
<n>:		"Вес точки"		
		Диапазон значений:	0 ≤ n ≤ 3	
		Размер шага:	0.0001	
		Действие:	n > 1	Кривая сильнее притягивается к контрольной точке.
			n < 1	Кривая слабее притягивается к контрольной точке.
Порядок сплайна (только В-сплайн):				
SD:		Стандартно используется полигон 3-го порядка. Но через программирование SD=2 возможно и использование полигона 2-го порядка.		
Расстояние между узловыми точками (только В-сплайн):				
PL :		Выполняется внутреннее подходящее вычисление расстояний между узловыми точками. Но СЧПУ может обрабатывать и заданные расстояния между узловыми точками, которые с помощью команды PL указываются как т.н. длине интервалов параметров.		
<значение>:		Длина интервала параметров		
		Диапазон значений:	Как размер перемещения	
Переходная характеристика в начале сплайна (только А- или С-сплайн):				
BAUTO:		Данные для переходной характеристики не задаются. Начало следует из положения первой точки.		
BNAT:		Изгиб ноль		
BTAN:		Тангенциальный переход к предшествующему кадру (позиция стирания)		
Переходная характеристика в конце сплайна (только А- или С-сплайн):				

5.2 Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

EAUTO:	Данные для переходной характеристики не задаются. Конец следует из положения последней точки.
ENAT:	Изгиб ноль
ETAN:	Тангенциальный переход к предшествующему кадру (позиция стирания)
	

Примечание

Программируемая переходная характеристика не влияет на В-сплайн. В-сплайн в стартовой и конечной точке всегда является тангенциальным к контрольному полигону.

Граничные условия

- Возможно использование коррекции радиуса инструмента.
- Контроль столкновений выполняется в проекции на плоскость.

Примеры

Пример 1: В-сплайн

Программный код 1 (все веса 1)

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
```

5.2 Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Программный код 1 (все веса 1)

```
N60 X40 Y45
N70 X50 Y0
```

Программный код 2 (различный вес)

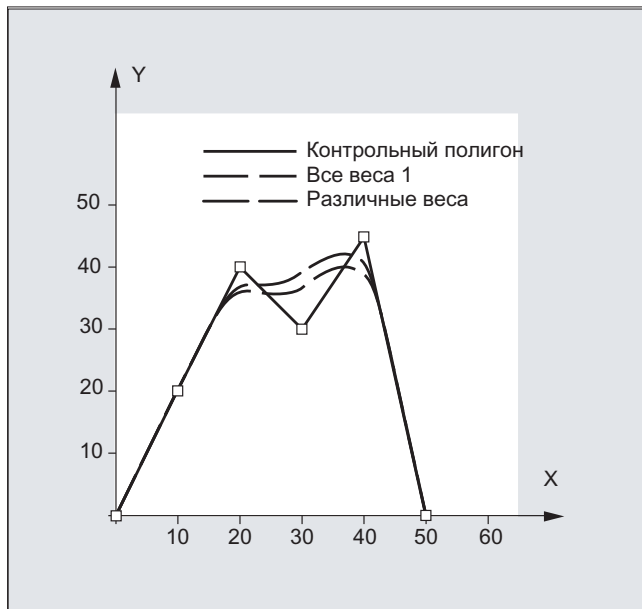
```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20 PW=2
N40 X20 Y40
N50 X30 Y30 PW=0.5
N60 X40 Y45
N70 X50 Y0
```

Программный код 3 (контрольный полигон)

Комментарий

```
N10 G1 X0 Y0 F300 G64
N20
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

; отсутствует



Пример 2: С-сплайн, изгиб ноль в начале и в конце

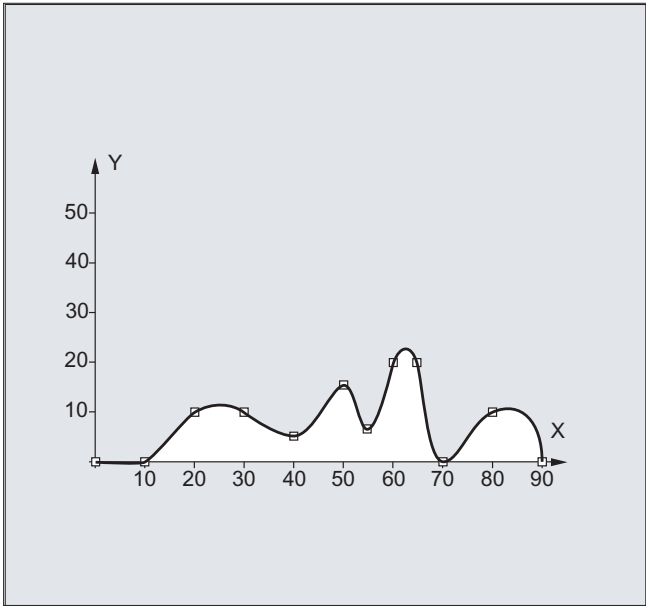
Программный код

```
N10 G1 X0 Y0 F300
```

5.2 Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Программный код

```
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
N40 X30
N50 X40 Y5
N60 X50 Y15
N70 X55 Y7
N80 X60 Y20
N90 X65 Y20
N100 X70 Y0
N110 X80 Y10
N120 X90 Y0
N130 M30
```



Пример 3: сплайн-интерполяция (А-сплайн) и трансформация координат (ROT)

Главная программа:

Программный код	Комментарий
N10 G00 X20 Y18 F300 G64	; Подвод в стартовую точку.
N20 ASPLINE	; Активировать тип интерполяции А-сплайна.
N30 KONTUR	; Первый вызов подпрограммы.
N40 ROT Z-45	; Трансформация координат: Поворот wCs на -45° вокруг оси Z.
N50 G00 X20 Y18	Подвод в стартовую точку контура.
N60 KONTUR	; Второй вызов подпрограммы.
N70 M30	; конец программы

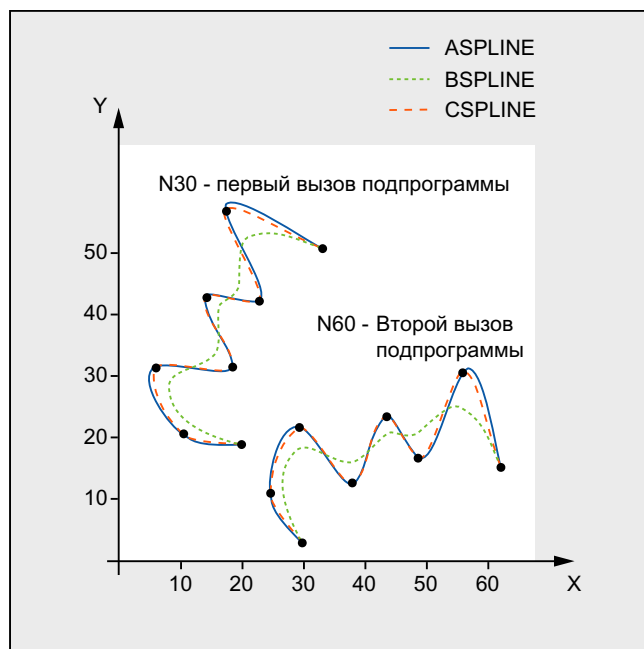
5.2 Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Подпрограмма "Kontur" (содержит координаты опорных точек):

Программный код

```
N10 X20 Y18
N20 X10 Y21
N30 X6 Y31
N40 X18 Y31
N50 X13 Y43
N60 X22 Y42
N70 X16 Y58
N80 X33 Y51
N90 M1
```

На рисунке ниже, наряду со сплайном, полученным из примера программирования (ASPLINE), содержатся и сплайны, которые получились бы при активации В- или С-сплайн-интерполяции (BSPLINE, CSPLINE):



Дополнительная информация

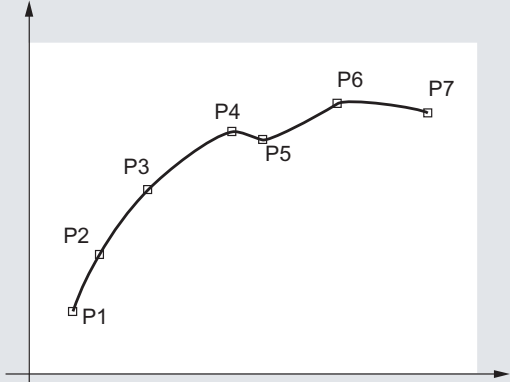
Преимущества сплайн-интерполяции

Благодаря использованию сплайн-интерполяции, в отличие от использования линейных кадров G01, могут быть достигнуты следующие преимущества:

- Уменьшение числа необходимых для описания контура кадров программы обработки детали
- Мягкий, щадящий для механики ход кривой на переходах между кадрами программы обработки детали

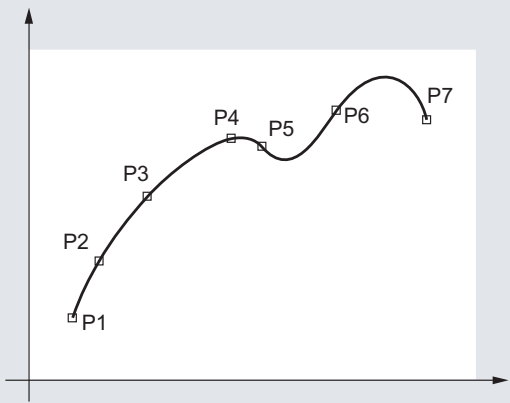
5.2 Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Свойства и использование различных типов сплайнов

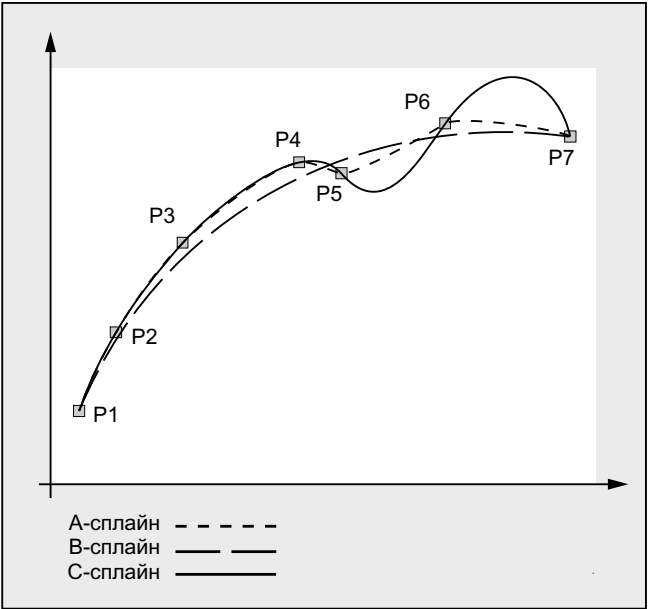
Тип сплайна	Свойства и использование
А-сплайн	А-сплайн (акима-сплайн)  P1 до P7: заданные координаты Свойства: • Проходит точно через заданные опорные точки. • Ход кривой стабильный по касательной и кривизне. • Практически не создает нежелательных колебаний. • Сфера влияния изменений опорных точек является локальной, т. е. изменение одной опорной точки сказывается только макс. на 6 соседних опорных точках. Использование: А-сплайн подходит прежде всего для интерполяции ходов кривой с большими изменениями наклона (например, ступенчатые ходы кривой).

Тип сплайна	Свойства и использование
В-сплайн	Свойства: • Проходит не через заданные опорные точки, а только вблизи. Кривая притягивается опорными точками. Посредством коэффициента эквивалентности опорных точек возможно дополнительное управление ходом кривой. • Ход кривой стабильный по касательной и кривизне. • Практически не создает нежелательных колебаний. • Сфера влияния изменений опорных точек является локальной, т. е. изменение одной опорной точки сказывается только макс. на 6 соседних опорных точках. Использование: Первоначально В-сплайн задумывался как интерфейс к системам CAD.

5.2 Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, PW, SD, PL)

Тип сплайна	Свойства и использование
С-сплайн	С-сплайн (кубический сплайн)  P1 до P7: заданные координаты Свойства: • Проходит точно через заданные опорные точки. • Ход кривой стабильный по касательной и кривизне. • Часто создает нежелательные колебания, особенно в местах с большими изменениями наклона. • Сфера влияния изменений опорных точек является глобальной, т. е. изменение одной опорной точки сказывается на всем ходе кривой. Использование: С-сплайн может успешно использоваться тогда, когда опорные точки лежат на аналитически известной кривой (окружность, парабола, гипербола)

Сравнение трех типов сплайнов в случае одинаковых опорных точек



Мин. число сплайн-кадров

G-команды ASPLINE, BSPLINE, CSPLINE соединяют конечные точки кадров сплайнами. Для этого на стадии предварительной обработки необходимо одновременное вычисление ряда кадров (конечных точек). Размер буфера для вычисления стандартно составляет 10 кадров. Не любая информация кадра является конечной точкой сплайна. Но СЧПУ из 10 кадров требуется определенное число кадров конечных точек сплайна:

Тип сплайна	Мин. число сплайн-кадров
А-сплайн:	Из каждых 10 кадров как минимум 4 должны быть сплайн-кадрами. Кадры комментариев и вычисления параметров при этом не учитываются.
В-сплайн:	Из каждых 10 кадров как минимум 6 должны быть сплайн-кадрами. Кадры комментариев и вычисления параметров при этом не учитываются.
С-сплайн:	Необходимое мин. число сплайн-кадров получается из следующей суммы: Значение из MD20160 \$MC_CUBIC_SPLINE_BLOCKS + 1 В MD20160 вносится число точек, через которые вычисляется сегмент сплайна. Стандартная настройка 8. Поэтому из каждых 10 кадров в стандартном случае как минимум 9 должны быть сплайн-кадрами.

Примечание

При значении, ниже допустимого, выводится ошибка, как и в том случае, когда участвующая в сплайне ось программируется как позиционирующая ось.

5.2 Сплайн-интерполяция (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Объединение коротких сплайн-кадров

При сплайн-интерполяции могут возникать короткие сплайн-кадры, которые приводят не к ненужному уменьшению скорости движения по траектории. С помощью функции "Соединение коротких сплайн-кадров" эти кадры могут быть соединены таким образом, что полученная длина кадра является достаточной и не приводит к уменьшению скорости движения по траектории.

Функция активируется через спец. для канала машинные данные:

MD20488 \$MC_SPLINE_MODE (настройка для сплайн-интерполяции)

Литература:

Описание функций "Основные функции"; Режим управления траекторией, точный останов, LookAhead (B1),
глава: Объединение коротких сплайн-кадров

5.3 Соединение сплайнов (SPLINEPATH)

Интерполирующие в соединении сплайнов оси выбираются с помощью команды SPLINEPATH. До восьми траекторных осей возможно при сплайн-интерполяции.

Примечание

Если SPLINEPATH не программируется явно, то первые три оси канала перемещаются как соединение сплайнов.

Синтаксис

Определение соединения сплайнов выполняется в отдельном кадре:

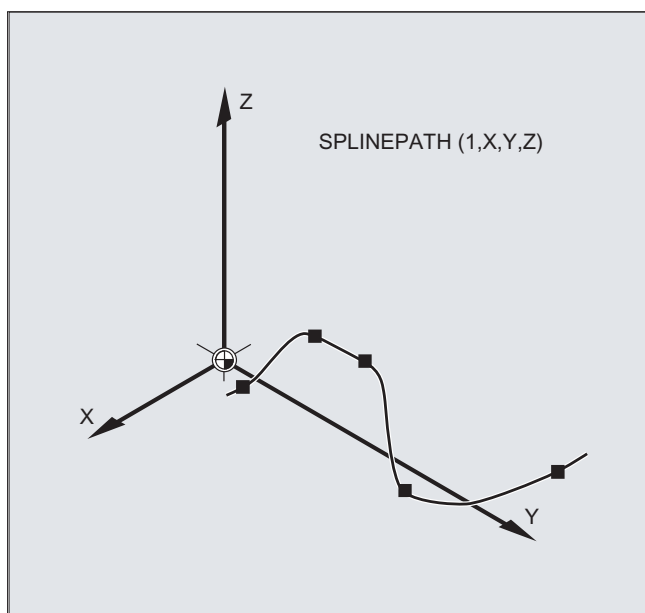
SPLINEPATH (n, X, Y, Z, ...)

Значение

SPLINEPATH:	Команда для определения соединения сплайнов
n:	=1 (фиксированное значение)
X, Y, Z, ... :	Идентификаторы интерполирующих в соединении сплайнов траекторных осей

Пример: Соединение сплайнов с тремя траекторными осями

Программный код	Комментарий
N10 G1 X10 Y20 Z30 A40 B50 F350	
N11 SPLINEPATH(1,X,Y,Z)	; Соединение сплайнов
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60	; C-сплайн
N14 X30 Y40 Z50 A60 B70	; Опорные точки
...	
N100 G1 X... Y...	; Отмена сплайн-интерполяции



5.4 Включение / выключение сжатия кадра ЧПУ (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)

Функции компрессора линейных кадров (и в зависимости от параметрирования также круговых кадров и/или кадров ускоренного хода) включаются / отключаются G-командами из группы G 30. Команды действуют модально.

Синтаксис

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

Объяснение

COMPON:	Включение функции компрессора COMPON
COMPCURV:	Включение функции компрессора COMPCURV
COMPCAD:	Включение функции компрессора COMPCAD
COMPSURF:	Включение функции компрессора COMPSURF
COMPOF:	Выключение текущей активной функции компрессора

Примечание

Для дополнительного улучшения качества поверхности можно использовать функцию перешлифовки G642 и ограничение рывка SOFT . Эти команды записываются в начале программы.

Пример: COMPCAD

Программный код	Комментарий
N10 G00 X30 Y6 Z40	
N20 G1 F10000 G642	; Включение: Функция перешлифовки G642
N30 SOFT	; Включение: Ограничение рывка SOFT
N40 COMPCAD	; Включение: Функция компрессора COMPCAD
N50 STOPFIFO	
N24050 Z32.499	; 1-й кадр перемещения
N24051 X41.365 Z32.500	; 2-й кадр перемещения
...	
N99999 X... Z...	; Последний кадр перемещения
COMPOF	; Функция компрессора выкл.
...	

5.5 Полиномиальная интерполяция (POLY, POLYPATH, PO, PL)

В прямом смысле в случае полиномиальной интерполяции (POLY) речь идет не о виде сплайн-интерполяции. В первую очередь она задумана как интерфейс для программирования созданных на внешних устройствах сплайн-кривых. При этом сегменты сплайна могут программироваться напрямую.

Такой тип интерполяции снимает с ЧПУ задачу по вычислению коэффициентов полиномов. Ее оптимальное использование возможно тогда, когда коэффициенты поступают напрямую от системы CAD или постпроцессора.

Синтаксис

Полином 3-го порядка:

POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n

Полиномы 5-го порядка и новый синтаксис полинома:

POLY X=PO(xe,a2,a3,a4,a5) Y=PO(ye,b2,b3,b4,b5) Z=PO(ze,c2,c3,c4,c5)

PL=n

POLYPATH ("AXES", "VECT")

Примечание

Сумма запрограммированных в одном кадре ЧПУ коэффициентов полиномов и осей не должна превышать макс. разрешенного числа осей на кадр.

Значение

POLY :	Включение полиномиальной интерполяции с помощью кадра с POLY.
POLYPATH :	Полиномиальная интерполяция с возможностью выбора для обеих групп осей AXIS или VECT
PO[идентификатор оси/переменная] :	Конечный точки и коэффициенты полиномов
X, Y, Z :	Идентификатор оси
xe, ye, ze :	Указание конечной позиции для соответствующей оси; диапазон значений как размер перемещения

5.5 Полиномиальная интерполяция (POLY, POLYPATH, PO, PL)

a2, a3, a4, a5:	Коэффициенты a ₂ , a ₃ , a ₄ , и a ₅ записываются с их значением; диапазон значений как размер перемещения. Соответствующий последний коэффициент не нужен, если он имеет значение ноль.
PL:	Длина интервала параметров, на котором определены полиномы (область определения функции f(p)). Интервал всегда начинается на 0, p может принимать значения от 0 до PL. Теоретический диапазон значений для PL: 0,0001 ... 99 999,9999 Примечание: Значение PL действует для кадра, в котором оно стоит. Если PL не запрограммировано, то действует PL=1.

Включение/выключение полиномиальной интерполяции

Полиномиальная интерполяция включается в программе обработки детали через G-команду POLY.

G-команда POLY вместе с G0, G1, G2, G3, ASPLINE, BSPLINE и CSPLINE относится к 1-й G-группе.

Оси, запрограммированные только с именем и конечной точкой (к примеру, X10), перемещаются линейно. Если таким образом запрограммированы все оси, то поведение СЧПУ как при G1.

Полиномиальная интерполяция снова не явно отключается через программирование другой команды 1-й G-группы (к примеру, G0, G1).

Коэффициент полинома

Значение PO (PO[]=) или ...=PO(...) указывает все коэффициенты полиномов для оси. Согласно порядку полинома несколько значений указываются разделенными запятой. Внутри одного кадра возможны различные порядки полинома для различных осей.

Подпрограмма POLYPATH

С POLYPATH(...) полиномиальная интерполяция может разрешаться выборочно для определенных групп осей:

Только траекторные и дополнительные оси:	POLYPATH ("AXES")
Только оси ориентации:	POLYPATH ("VECT")
(при перемещении с трансформацией ориентации)	

Соответствующие не разрешенные оси перемещаются линейно.

Стандартно полиномиальная интерполяция разрешена для обеих групп осей.

5.5 Полиномиальная интерполяция (POLY, POLYPATH, PO, PL)

Через программирование без параметра POLYPATH () полиномиальная интерполяция деактивируется для всех осей.

Пример

Программный код	Комментарий
N10 G1 X... Y... Z... F600	
N11 POLY PO[X]=(1,2.5,0.7) PO[Y]=(0.3,1,3.2) PL=1.5	; Полиномиальная интерполяция вкл
N12 PO[X]=(0,2.5,1.7) PO[Y]=(2.3,1.7) PL=3	
...	
N20 M8 H126 ...	
N25 X70 PO[Y]=(9.3,1,7.67) PL=5	; Смешанные данные для осей
N27 PO[X]=(10,2.5) PO[Y]=(2.3)	; PL не запрограммирован; действует PL=1
N30 G1 X... Y... Z.	; Полиномиальная интерполяция выкл
...	

Пример: Новый синтаксис полинома

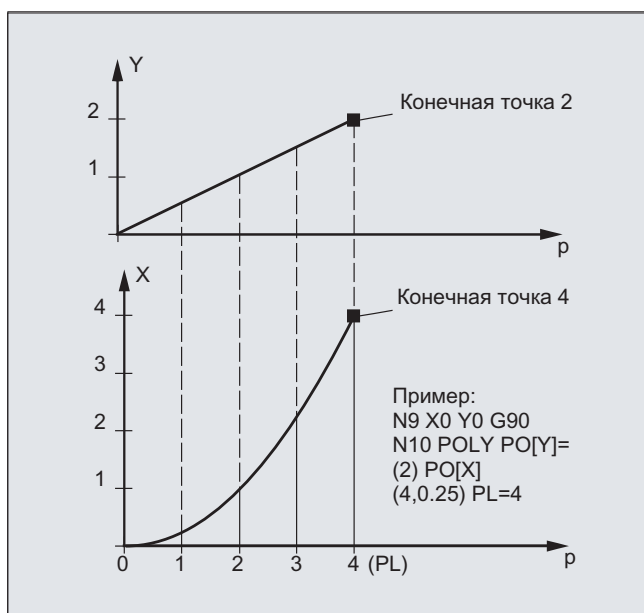
Продолжающий действовать синтаксис полинома	Новый синтаксис полинома
PO[идентификатор оси]=(.. , ..)	Идентификатор оси=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[переменная]=IC(.. , ..)	Переменная=PO IC(.. , ..)

Пример: Кривая в плоскости X/Y

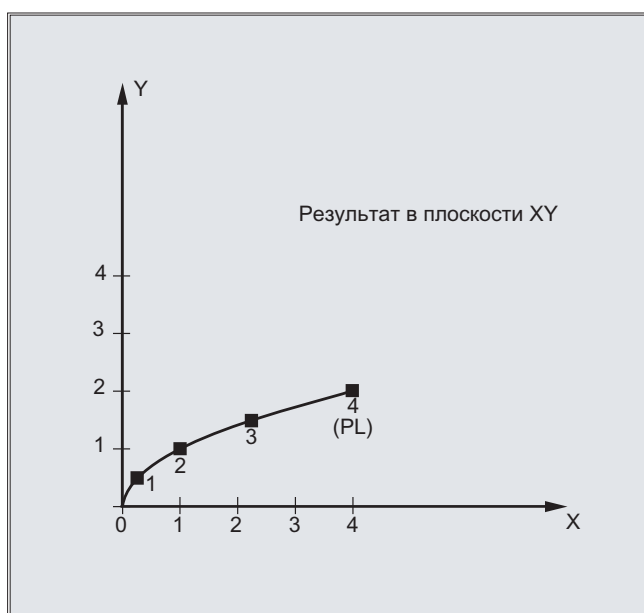
Программирование

Программный код
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4

Ход кривых X(p) и Y(p)



Ход кривой в плоскости XY



Описание

В общей форме функция полинома выглядит следующим образом:

$$f(p) = a_0 + a_1 p + a_2 p^2 + \dots + a_n p^n$$

где: a_i : постоянные коэффициенты ($i = 0, 1, \dots, n$)
 p : параметры

В СЧПУ можно программировать полиномы не выше 5-го порядка:

$$f(p) = a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$$

Посредством присваивания конкретных значений коэффициентам можно создавать различные ходы кривой, например, линейные, параболические, степенные функции.

Прямая создается через $a_2 = a_3 = a_4 = a_5 = 0$:

$$f(p) = a_0 + a_1p$$

Дальше действует:

a_0 : Позиция оси на конце предшествующего кадра

$p = PL$

$$a_1 = (x_E - a_0 - a_2 \cdot p^2 - a_3 \cdot p^3) / p$$

Можно программировать полиномы **без** активации полиномиальной интерполяции через G-команду POLY. Но в этом случае выполняется интерполяция не запрограммированных полиномов, а осуществляется линейный подвод к запрограммированным конечным точкам осей (G1). Только после явной активации полиномиальной интерполяции в программе обработки детали (POLY) запрограммированные полиномы начинают перемещаться как таковые.

Особенность: Полином-знаменатель

Для геометрических осей с помощью $PO[] = (...)$ без указания имени оси также можно запрограммировать общий полином-знаменатель, т. е. движение геометрических осей интерполируется как частное двух полиномов.

Таким образом, могут быть точно представлены, например, конические сечения (круг, эллипс, парабола, гипербола).

Пример:

Программный код	Комментарий
POLY G90 X10 Y0 F100	; Гео-оси перемещаются линейно на позицию X10, Y0.
PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)	; Гео-оси перемещаются по четверти круга на X0 Y10.

Постоянный коэффициент (a_0) полинома-знаменателя всегда принимается за 1.

Запрограммированная конечная точка не зависит от G90 / G91.

Из запрограммированных значений $X(p)$ и $Y(p)$ вычисляются как:

$$X(p) = (10 - 10 \cdot p^2) / (1 + p^2)$$

$$Y(p) = 20 \cdot p / (1 + p^2)$$

$$\text{где } 0 \leq p \leq 1$$

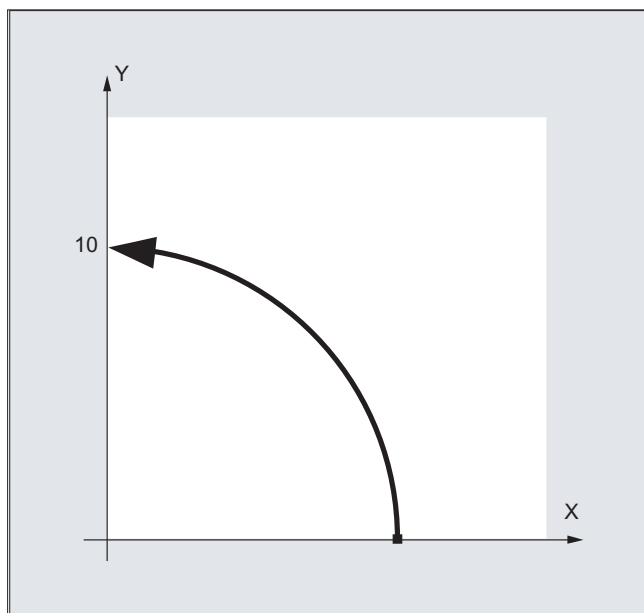
5.5 Полиномиальная интерполяция (POLY, POLYPATH, PO, PL)

На основе запрограммированных начальных точек, конечных точек, коэффициента a_2 и $PL=1$ получаются следующие промежуточные результаты:

$$\text{Числитель (X)} = 10 + 0 \cdot p - 10 \cdot p^2$$

$$\text{Числитель (Y)} = 0 + 20 \cdot p + 0 \cdot p^2$$

$$\text{Знаменатель} = 1 + p^2$$



При включенной полиномиальной интерполяции программирование полинома-знаменателя с нулями внутри интервала $[0, PL]$ отклоняется с ошибкой. Полином-знаменатель не влияет на движение дополнительных осей.

Примечание

Коррекция радиуса инструмента для полиномиальной интерполяции может включаться с G41, G42 и использоваться как для линейной или круговой интерполяции.

5.6 Устанавливаемое соотношение траекторий (SPATH, UPATH)

При полиномиальной интерполяции (POLY, ASPLINE, BSPLINE, CSPLINE, COMPON, COMPCURV) позиции траекторных осей i задаются через полиномы $p_i(U)$. Параметр кривой U при этом перемещается внутри кадра ЧПУ от 0 до 1.

С помощью FGROUР выбираются оси (оси FGROUР), к которым относится подача по траектории F . Но интерполяция с постоянной скоростью на пути S осей FGROUР означает при полиномиальной интерполяции, как правило, не постоянное изменение параметра кривой U . Поэтому для осей, не содержащихся в FGROUР, возможен выбор из двух способов следования этих осей за осями FGROUР:

- Синхронно ходу траектории S (SPATH)
- Синхронно параметру кривой U (UPATH)

Синтаксис

SPATH
UPATH

Значение

SPATH:	Не содержащиеся в FGROUР оси перемещаются относительно хода траектории S
UPATH:	Не содержащиеся в FGROUР оси перемещаются относительно параметра кривой U

Примечание

UPATH и SPATH определяют и связь полинома слова F (FPOLY, FCUB, FLIN) с движением по траектории.

Граничные условия

SPATH или UPATH не имеют значения в следующих случаях:

- линейная интерполяция ($G1$)
- круговая интерполяция ($G2, G3$)
- кадры резьбы ($G33, G34, G35, G33x, G63$)
- все траекторные оси имеются в FGROUР

Пример

Пример ниже показывает разницу между двумя типами управления движением.

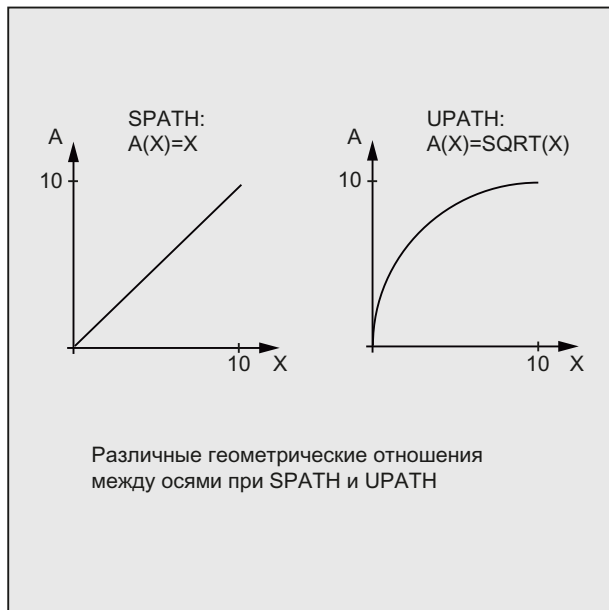
Программный код
N10 FGROUР(X, Y, Z)
N15 G1 X0 A0 F1000 SPATH ; SPATH
N20 POLY PO[X]=(10,10) A10

5.6 Устанавливаемое соотношение траекторий (SPATH, UPATH)

Программный код

```
N10 FGROUP (X,Y,Z)
N15 G1 X0 A0 F1000 UPATH ; UPATH
N20 POLY PO[X]=(10,10) A10
```

В обоих сегментах программы ход S осей FGROUP в N20 зависит от квадрата параметра кривой U. Поэтому вдоль хода X получаются различные позиции синхронной оси A, в зависимости от того, активна ли SPATH или UPATH.



Дополнительная информация

Поведение СЧПУ при Reset и машинные/опциональные данные

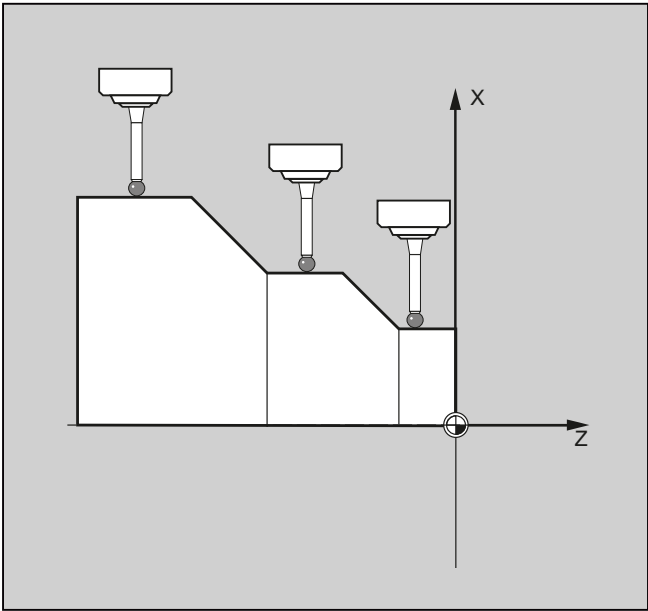
После Reset действует определенный посредством MD20150 \$MC_GCODE_RESET_VALUES[44] G-команда (45-я G-группа).

Значение по умолчанию для типа перешлифовки определяется посредством MD20150 \$MC_GCODE_RESET_VALUES[9] (10-я G-группа).

Спец. для оси машинные данные MD33100 \$MA_COMPRESS_POS_TOL[<n>] имеют расширенное значение: они содержат допуски для функции компрессора и для перешлифовки с G642.

5.7 Измерение с помощью контактного щупа (MEAS, MEAW)

С помощью функции "Измерение с контактным щупом" выполняется подвод к фактическим позициям на детали и при контактном фронте измерительного щупа для всех запрограммированных в кадре измерения осей измеряются позиции и записываются для каждой оси в соответствующую ячейку памяти.



Для программирования функции предлагаются два следующих постоянных адреса:

- MEAS
С помощью MEAS удаляется оставшийся путь между фактической и заданной позицией.
- MEAW
Для задач измерения, в которых в любом случае необходим подвод к запрограммированной позиции, используется MEAW.

MEAS и MEAW действуют покадрово и программируются вместе с операторами движения. Подача и тип интерполяции (G0, G1, ...), как и число осей, при этом должны быть согласованы с соответствующей задачей измерения.

Синтаксис

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

Значение

MEAS:	Команда: Измерение со стиранием остатка пути	
	Активность:	покадрово
MEAW:	Команда: Измерение без стирания остатка пути	
	Активность:	покадрово

5.7 Измерение с помощью контактного щупа (MEAS, MEAW)

<TE>:	Пусковое событие для запуска измерения		
	Тип:	INT	
	Диапазон значений:	-2, -1, 1, 2	
	Значение:		
	(+)1	передний фронт измерительного щупа 1 (на входе измерительной системы 1)	
	-1	задний фронт измерительного щупа 1 (на входе измерительной системы 1)	
	(+)2	передний фронт измерительного щупа 2 (на входе измерительной системы 2)	
	-2	задний фронт измерительного щупа 2 (на входе измерительной системы 2)	
	Примечание: Существует макс. 2 измерительных щупа (в зависимости от конфигурации).		
G . . . :	Тип интерполяции, к примеру, G0, G1, G2 или G3		
X . . . Y . . . Z . . . :	Конечные точки в декартовых координатах		

Пример

Программный код	Комментарий
N10 MEAS=1 G1 F1000 X100 Y730 Z40	; Кадр измерения со щупом первого измерительного входа и линейная интерполяция. Остановка предварительной обработки создается автоматически.
...	

Дополнительная информация

Состояние задания измерения

Если в программе необходимо проанализировать, были ли контакт щупа или нет, можно опросить переменную состояния \$AC_MEA[<n>] (<n> = номер измерительного щупа):

Значение	Объяснение
0	Задание измерения не выполнено
1	Задание измерения успешно завершено (контакт измерительного щупа)

Примечание

Если измерительный щуп отклоняется в программе, то переменная устанавливается на 1. При старте кадра измерения переменная автоматически устанавливается на начальное состояние щупа.

Запись измеренных значений

Регистрируются позиции всех перемещенных траекторных и позиционирующих осей кадра (макс. число осей в зависимости от конфигурации СЧПУ). При MEAS после контакта щупа происходит определенное притормаживание.

Примечание

Если в кадре измерения запрограммирована гео-ось, то сохраняются измеренные значения для всех текущих гео-осей.

Если в кадре измерения запрограммирована участвующая в трансформации ось, то сохраняются измеренные значения всех участвующих в этой трансформации осей.

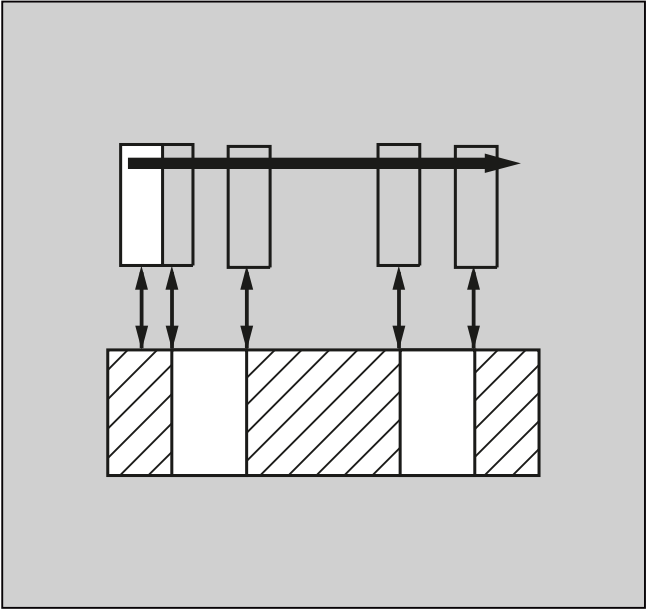
Чтение результатов измерения

Результаты измерения для обработанных с помощью щупа осей могут быть считаны через следующие системные переменные:

- \$AA_MM[<ось>]
Результаты измерения в системе координат станка
- \$AA_MW[<ось>]
Результаты измерения в системе координат детали

5.8 Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):

Для спец. для оси измерения могут использоваться несколько измерительных щупов и несколько измерительных систем.



Для программирования функции предлагаются кодовые слова MEASA, MEAWA и MEAC.

С MEASA, MEAWA для соответствующей запрограммированной оси регистрируется до четырех измеренных значений на измерение и сохраняется согласно пусковому событию в системных переменных.

Непрерывные задания измерения могут выполняться с MEAC. В этом случае результаты измерения сохраняются в переменных FIFO.

Синтаксис

```
MEASA [<ось>]=(<режим>, <TE1>, ..., <TE4>)  
MEAWA [<ось>]=(<режим>, <TE1>, ..., <TE4>)  
MEAC [<ось>]=(<режим>, <память измерений>, <TE1>, ..., <TE4>)
```

Примечание

MEASA и MEAWAдействуют покадрово и могут быть запрограммированы вместе в одном кадре. Если же напротив MEASA/MEAWA вместе с MEAS/MEAW программируются в одном кадре, то следует сообщение об ошибке.

Объяснение

MEASA:	Кодовое слово: Спец. для оси измерение со стиранием остатка пути	
	Активность:	покадрово

5.8 Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):

MEAWA:	Кодовое слово: Спец. для оси измерение без стирания остатка пути	
	Активность:	покадрово
MEAC:	Кодовое слово: Спец. для оси непрерывное измерение без стирания остатка пути	
	Активность:	покадрово
<ось>:	Имя используемой для измерения оси канала	
<режим>:	Двухзначное число для указания режима работы (режим измерения и измерительная система)	
	Первый десятичный разряд (режим измерения):	
	0	Отменить задание измерения.
	1	До 4 различных одновременно активируемых пусковых событий.
	2	До 4 последовательно активируемых пусковых событий.
	3	До 4 последовательно активируемых пусковых событий, но без контроля пускового события 1 при запуске (аварийные сообщения 21700/21703 подавляются).
	Примечание: Этот режим при MEAC невозможен.	
	Второй десятичный разряд (измерительная система):	
	0 (или данные отсутствуют)	Активная измерительная система
	1	Измерительная система 1
	2	Измерительная система 2
	3	Обе измерительные системы
<ТЕ>:	Пусковое событие для запуска измерения	
	Тип:	INT
	Диапазон значений:	-2, -1, 1, 2
	Объяснение:	
	(+)1	передний фронт измерительного щупа 1
	-1	задний фронт измерительного щупа 1
	(+)2	передний фронт измерительного щупа 2
	-2	задний фронт измерительного щупа 2
<память измерений>:	Номер FIFO (динамическая память)	

Примеры

Пример 1: Спец. для оси измерение со стиранием остатка пути в режиме 1 (обработка во временной последовательности)

а) С 1 измерительной системой

Программный код	Комментарий
... N100 MEASA[X]=(1,1,-1) G01 X100 F100	; Измерение в режиме 1 с активной измерительной системой. Ожидание сигнала измерения с передним/задним фронтом измерительного щупа 1 на пути перемещения к X=100.

5.8 Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):

Программный код	Комментарий
N110 IF \$AC_MEA[1]==FALSE GOTOF ENDE	; Проверить успех измерения.
N120 R10=\$AA_MM1[X]	; Сохранить относящееся к первому запрограммированному пусковому событию (передний фронт) измеренное значение.
N130 R11=\$AA_MM2[X]	; Сохранить относящееся ко второму запрограммированному пусковому событию (задний фронт) измеренное значение.
N140 ENDE:	

b) С 2 измерительными системами

Программный код	Комментарий
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; Измерение в режиме 1 с обеими измерительными системами. Ожидание сигнала измерения с передним/задним фронтом измерительного щупа 1 на пути перемещения к X=100.
N210 IF \$AC_MEA[1]==FALSE GOTOF ENDE	; Проверить успех измерения.
N220 R10=\$AA_MM1[X]	; Сохранить измеренное значение измерительной системы 1 при переднем фронте.
N230 R11=\$AA_MM2[X]	; Сохранить измеренное значение измерительной системы 2 при переднем фронте.
N240 R12=\$AA_MM3[X]	; Сохранить измеренное значение измерительной системы 1 при заднем фронте.
N250 R13=\$AA_MM4[X]	; Сохранить измеренное значение измерительной системы 2 при заднем фронте.
N260 ENDE:	

Пример 2: Спец. для оси измерение со стиранием остатка пути в режиме 2 (обработка в запрограммированной последовательности)

Программный код	Комментарий
...	
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	; Измерение в режиме 2 с активной измерительной системой. Ожидание сигнала измерения в последовательности передний фронт измерительного щупа 1, задний фронт измерительного щупа 1, передний фронт измерительного щупа 2, задний фронт измерительного щупа 2 на пути перемещения к X=100.
N110 IF \$AC_MEA[1]==FALSE GOTOF MESSTASTER2	; Проверить успех измерения с измерительным щупом 1.
N120 R10=\$AA_MM1[X]	; Сохранить относящееся к первому запрограммированному пусковому событию (передний фронт щупа 1) измеренное значение.
N130 R11=\$AA_MM2[X]	; Сохранить относящееся ко второму запрограммированному пусковому событию (передний фронт щупа 1) измеренное значение.
N140 MESSTASTER2:	

5.8 Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):

Программный код	Комментарий
N150 IF \$AC_MEA[2]==FALSE GOTOF ENDE	; Проверить успех измерения с измерительным щупом 2.
N160 R12=\$AA_MM3[X]	; Сохранить относящееся к третьему запрограммированному пусковому событию (передний фронт щупа 2) измеренное значение.
N170 R13=\$AA_MM4[X]	; Сохранить относящееся к четвертому запрограммированному пусковому событию (передний фронт щупа 2) измеренное значение.
N180 ENDE:	

Пример 3: Спец. для оси непрерывное измерение в режиме 1 (обработка во временной последовательности)

а) Измерение до 100 значений

Программный код	Комментарий
...	
N110 DEF REAL MESSWERT[100]	
N120 DEF INT Schleife=0	
N130 MEAC[X]=(1,1,-1) G01 X1000 F100	; Измерение в режиме 1 с активной измерительной системой, сохранение измеренных значений в \$AC_FIFO1, ожидание сигнала измерения с задним фронтом измерительного щупа 1 на пути перемещения к X=1000.
N135 STOPRE	
N140 MEAC[X]=(0)	; Отменить измерение после достижения позиции оси.
N150 R1=\$AC_FIFO1[4]	; Сохранить число полученных измеренных значений в параметре R1.
N160 FOR Schleife=0 TO R1-1	
N170 MESSWERT[Schleife]=\$AC_FIFO1[0]	; Выгрузить и сохранить измеренные значения из \$AC_FIFO1.
N180 ENDFOR	

б) Измерение со стиранием остатка пути после 10 измеренных значений

Программный код	Комментарий
...	
N10 WHEN \$AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; Стирание остаточного пути.
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC[X]=(0)	
N40 R1=\$AC_FIFO1[4]	; Число измеренных значений.
...	

5.8 Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):

с) Измерение заднего/переднего профиля зуба с 2 измерительными щупами

Программный код	Комментарий
...	
N110 DEF REAL MESSWERT[16]	
N120 DEF INT Schleife=0	
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100	; Измерение в режиме 1 с активной измерительной системой, сохранение измеренных значений в \$AC_FIFO1, ожидание сигнала измерения в последовательности заднего фронта измерительного щупа 1, переднего фронта измерительного щупа 2, на пути перемещения к X=100.
N140 STOPRE	; Остановка предварительной обработки
N150 MEAC[X]=(0)	; Отменить измерение после достижения позиции оси.
N160 R1=\$AC_FIFO1[4]	; Сохранить число полученных измеренных значений в параметре R1.
N170 FOR Schleife=0 TO R1-1	
N180 MESSWERT[Schleife]=\$AC_FIFO1[0]	; Выгрузить и сохранить измеренные значения из \$AC_FIFO1.
N190 ENDFOR	

Дополнительная информация

Задание измерения

Программирование может осуществляться в программе обработки детали или из синхронного действия (см. главу "Синхронные действия (Страница 641)"). Для каждой оси в один и тот же момент времени может быть активно только одно задание измерения.

Примечание

Подача должна быть согласована с соответствующим заданием измерения.

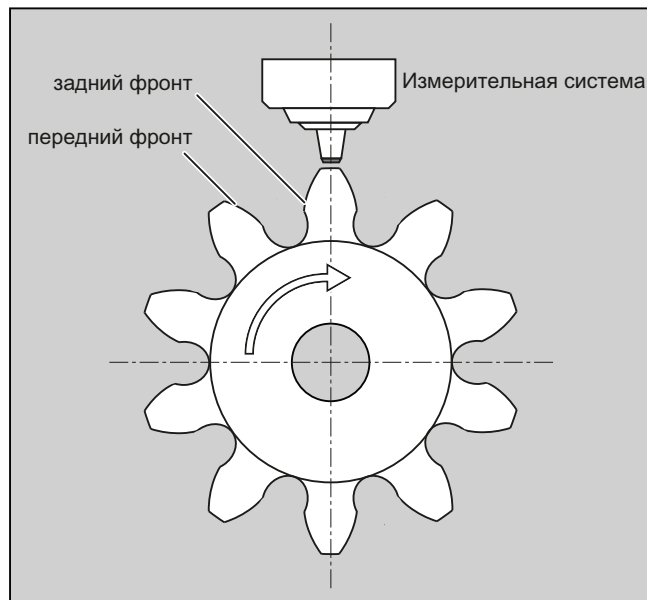
Для MEASA и MEAWA правильные результаты могут быть обеспечены только для подач, при которых возникает не более одного одинакового и не более 4 различных пусковых события на такт регулятора положения.

При непрерывном измерении с MEAC отношение между тактом интерполятора и тактом регулятора положения не может быть больше, чем 1:8.

Пусковое событие

Пусковое событие состоит из номера измерительного щупа и критерия запуска (передний или задний фронт) сигнала измерения.

5.8 Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):



Для каждого измерения может быть обработано до 4 пусковых событий сработавших измерительных щупов соответственно, т. е. до двух измерительных щупов с двумя измерительными фронтами каждый. Последовательность обработки, а также макс. число пусковых событий, зависят при этом от выбранного режима.

Примечание

Для режима измерения 1 действует: Одно и то же пусковое событие может быть запрограммировано в одном задании измерения только один раз!

В случае MEAC количество измеренных значений на каждое пусковое событие можно увеличить телеграммой PROFIBUS 395, в общей сложности, до 8 измеренных значений при переднем фронте и до 8 при заднем фронте на каждое пусковое событие и такт регулятора.

- Один измерительный щуп: 8 измеренных значений при переднем фронте и 8 при заднем фронте
- Два измерительных щупа: 4 измеренных значения при переднем фронте и 4 при заднем фронте на каждый измерительный щуп

Благодаря этому при использовании телеграммы PROFIBUS 395 можно реализовать более высокие значения подачи или скорости вращения.

Литература:

Описание функций "Дополнительные функции"; Измерение (M5), глава: Осевое измерение

Режим работы

С помощью первой цифры (второй десятичный разряд) режима выбирается желаемая измерительная система. Если имеется только одна измерительная система, но запрограммирована вторая, то автоматически используется имеющаяся.

5.8 Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):

С помощью второй цифры (первый десятичный разряд) выбирается желаемый режим измерения. Тем самым процесс измерения адаптируется к возможностям соответствующей СЧПУ:

- **Режим 1**
Обработка пусковых событий выполняется во временной последовательности их возникновения. В этом режиме при использовании шести осевых модулей может быть запрограммировано только одно пусковое событие или при указании нескольких пусковых событий происходит автоматический переход в режим 2 (без сообщения).
- **Режим 2**
Обработка пусковых событий выполняется в запрограммированной последовательности.
- **Режим 3**
Обработка пусковых событий выполняется в запрограммированной последовательности, но без контроля пускового события 1 при START.

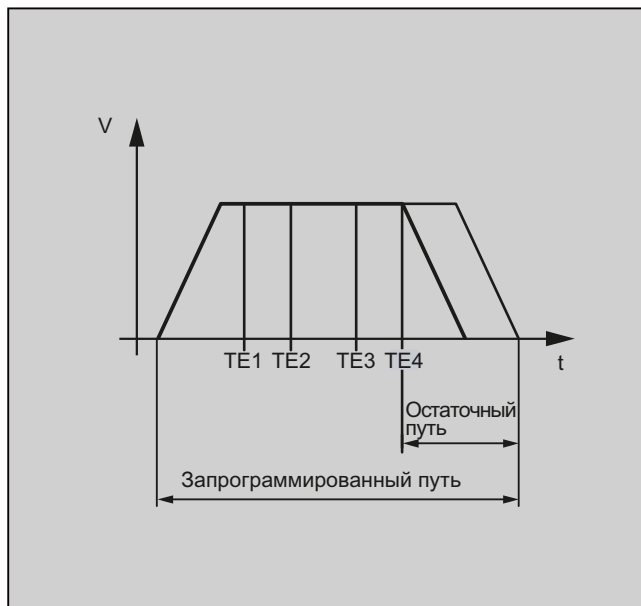
Примечание

При использовании 2 измерительных систем может быть запрограммировано только два пусковых события.

Измерение с и без стирания остатка пути

При программировании MEASA стирание остатка пути выполняется только после регистрации всех требуемых измеренных значений.

Для специальных задач измерения, в которых в любом случае необходим подвод к запрограммированной позиции, используется MEAWA .



Примечание

MEASA не может быть запрограммирована в синхронных действиях. В качестве эквивалента MEAWA плюс стирание остатка пути можно запрограммировать как синхронное действие.

Если задание измерения запускается с MEAWA из синхронных действий, то измеренные значения доступны только в системе координат станка.

Результаты измерения для MEASA, MEAWA

Результаты измерения доступны в следующих системных переменных:

- В системе координат станка:

\$AA_MM1 [<ось>]	Измеренное значение запрограммированной измерительной системы для пускового события 1
...	...
\$AA_MM4 [<ось>]	Измеренное значение запрограммированной измерительной системы для пускового события 4

- В системе координат детали:

\$AA_MW1 [<ось>]	Измеренное значение запрограммированной измерительной системы для пускового события 1
...	...
\$AA_MW4 [<ось>]	Измеренное значение запрограммированной измерительной системы для пускового события 4

Гео-оси / трансформации

Если осевое измерение должно быть запущено для гео-оси, то такое же задание измерения должно быть явно запрограммировано для всех оставшихся гео-осей. Это же относится и к осям, участвующим в трансформации.

Примеры:

```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

или

```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

Задание измерения с 2 измерительными системами

Если задание измерения выполняется с двумя измерительными системами, то каждое из двух возможных пусковых событий регистрируются обеими измерительными

5.8 Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):

системами соответствующей оси. Тем самым значение зарезервированных переменных задано:

\$AA_MM1 [<ось>]	или	\$AA_MW1 [<ось>]	Измеренное значение измерительной системы 1 при пусковом событии 1
\$AA_MM2 [<ось>]	или	\$AA_MW2 [<ось>]	Измеренное значение измерительной системы 2 при пусковом событии 1
\$AA_MM3 [<ось>]	или	\$AA_MW3 [<ось>]	Измеренное значение измерительной системы 1 при пусковом событии 2
\$AA_MM4 [<ось>]	или	\$AA_MW4 [<ось>]	Измеренное значение измерительной системы 2 при пусковом событии 2

Системные переменные

Состояние измерительного щупа доступно в следующих системных переменных:

\$A_PROBE[<n>]

Значение	Объяснение
1	Измерительный щуп отклонен
0	Измерительный щуп не отклонен

Ограничение измерительного щупа доступно в следующих системных переменных:

\$A_PROBE_LIMITED[<n>]

Значение	Объяснение
1	Ограничение измерительного щупа активно
0	Ограничение измерительного щупа не активно

<n> = измерительный щуп

Литература:

Справочник по параметрированию "Системные переменные"

Состояние задания измерения для MEASA, MEAWA

Если в программе необходим анализ, то состояние задания измерения может быть запрошено через \$AC_MEA[<n>], где <n> = номер измерительного щупа. Как только все

5.8 Спец. для оси измерение (MEASA, MEAWA, MEAC) (опция):

запрограммированные в одном кадре пусковые события измерительных щупов <n> последовали, эта переменная выводит значение 1. В ином случае значение 0.

Примечание

Если измерение запускается из синхронных действий, то \$AC_MEA более не обновляется. В этом случае необходимо опросить новый интерфейсный сигнал PLC DB31, ... DBX62.3 или равнозначную переменную \$AA_MEAАСТ[<ось>].

Объяснение:

\$AA_MEAАСТ==1: Измерение активно

\$AA_MEAАСТ==0: Измерение не активно

Непрерывное измерение (MEAC)

В случае MEAC измеренные значения доступны в системе координат станка и помещаются в указанную память FIFO[n] (динамическая память). Если для измерения сконфигурировано два измерительных щупа, то измеренные значения второго щупа отдельно помещаются в дополнительно сконфигурированную для этого (настройка через MD) память FIFO[n+1].

Память FIFO это динамическая память, в которой измеренные значения по динамическому принципу вносятся в переменную \$AC_FIFO, см. главу "Синхронные действия (Страница 641)".

Примечание

Содержание FIFO может быть выгружено из динамической памяти только один раз. Для многократного использования измеренных данных они должны быть промежуточно сохранены в данных пользователя.

При превышении числом измеренных значений макс. установленного в машинных данных для памяти FIFO числа, измерение автоматически завершается.

Бесконечное измерение может быть реализовано посредством циклической выгрузки измеренных значений. При этом выгрузка должна осуществляться как минимум также часто, как поступление новых измеренных значений.

Литература:

- Описание функций "Синхронные действия"; Подробное описание, глава: Параметры (\$AC_FIFO)
- Описание функций "Дополнительные функции"; Измерение (M5), глава: Осевое измерение

Защита от ошибок программирования

Следующие ошибочные программирования определяются и индицируются с ошибкой:

- MEASA/MEAWA вместе с MEAS/MEAW запрограммированы в одном кадре.

Пример:

```
N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
```

- MEASA/MEAWA с числом параметров <2 или >5

Пример:

```
N01 MEAWA[X]=(1) G01 F100 POS[X]=100
```

- MEASA/MEAWA с пусковым событием, отличным от 1/ -1/ 2/ -2

Пример:

```
N01 MEASA[B]=(1,1,3) B100
```

- MEASA/MEAWA с неправильным режимом

Пример:

```
N01 MEAWA[B]=(4,1) B100
```

- MEASA/MEAWA с дважды запрограммированным пусковым событием

Пример:

```
N01 MEASA[B]=(1,1,-1,2,-1) B100
```

- MEASA/MEAWA и отсутствующая гео-ось

Пример:

```
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;ГЕО-ось X/  
Y/Z
```

- Несогласованное задание измерения для гео-осей

Пример:

```
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50  
F100
```


5.9 Специальные функции для OEM-пользователя (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

Адреса OEM

Значение адресов OEM определяет пользователь OEM. Функция реализуется через компилированные циклы. 5 адресов OEM зарезервировано (OMA1 ... OMA5). Идентификаторы адресов настраиваемые. Адреса OEM разрешены в каждом кадре.

Зарезервированные вызовы G-команд:

Для пользователя OEM зарезервированы следующие вызовы G-команд:

- OEMIPO1, OEMIPO2 (из группы G 1)
- G810 ... G819 (G-группа 31)
- G820 ... G829 (G-группа 32)

Функция реализуется через компилированные циклы.

Функции и подпрограммы

Дополнительно пользователи OEM могут создавать и предопределенные функции и подпрограммы с передачей параметров.

Примечание

Симуляция детали

При симуляции детали до ПО 4.4 не поддерживаются, от ПО 4.4 поддерживаются только некоторые компилируемые циклы (CC).

Поэтому языковые команды в программе обработки детали из не поддерживаемых компилируемых циклов (OMA1 ... OMA5, OEMIPO1/2, G810 ... G829, некоторые процессы и функции) приводят без индивидуальной обработки с сигнализации ошибки и отмене симуляции.

Решение: В индивидуальном порядке обработать отсутствующие CC-спец. языковые элементы в программе обработки детали (\$P_SIM-запрос).

Пример:

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ;при симуляции CC не активен
ELSE
N5 X300 OMA1=10
ENDIF
```

5.10 Уменьшение подачи с замедлением на углах (FENDNORM, G62, G621)

При автоматическом замедлении на углах подача уменьшается колоколообразно непосредственно перед соответствующим углом. Кроме этого, объем релевантных для обработки параметров инструмента может быть спараметрирован через установочные данные. Это:

- Начало и конец уменьшения подачи
- Процентовка, с которой уменьшается подача
- Определение релевантного угла

В качестве релевантных углов учитываются такие углы, внутренний угол которых меньше, чем спараметрированный через установочные данные угол.

При значении по умолчанию FENDNORM функция автоматической угловой процентовки отключается.

Литература:

/FBFA/ Описание функций "Диалекты ISO"

Синтаксис

FENDNORM

G62 G41

G621

Значение

FENDNORM:	Автоматическое замедление на углах выкл
G62:	Замедление на внутренних углах при активной коррекции радиуса инструмента
G621:	Замедление на всех углах при активной коррекции радиуса инструмента

G62 действует только на внутренних углах с

- активной коррекцией радиуса инструмента G41, G42 и
- активного режима управления траекторией G64, G641

Подвод к соответствующему углу выполняется с уменьшенной подачей, которая следует из:

$F * (\text{процентовка для уменьшения подачи}) * \text{процентовка подачи}$

Макс. возможное уменьшение подачи достигается тогда, когда инструмент, относительно траектории центра, должен выполнить изменение направления в соответствующем углу.

G621 действует аналогично G62 на каждом углу, определенных через FGROUP осей.

5.11 Программируемый критерий окончания движения (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

5.11 Программируемый критерий окончания движения (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Подобно критерию смены кадра при траекторной интерполяции (G601, G602 и G603), для командных осей/осей PLC в программе обработки детали или в синхронных действиях можно запрограммировать критерий окончания движения для интерполяции отдельной оси.

В зависимости от того, какой критерий окончания движения установлен, кадры программы обработки детали или кадры технологического цикла с движениями отдельных осей завершаются с различной скоростью. Это же относится и к PLC через FC15/16/18.

Синтаксис

```

FINEA [<ось>]
COARSEA [<ось>]
IPOENDA [<ось>]
IPOBRKA (<ось>[,<момент времени>])
ADISPOSA (<ось>[,<режим>,<размер окна>])
    
```

Значение

FINEA:	Критерий окончания движения: "Точный останов точный"		
	Активность:	модально	
COARSEA:	Критерий окончания движения: "Точный останов грубый"		
	Активность:	модально	
IPOENDA:	Критерий окончания движения:"Стоп интерполятора"		
	Активность:	модально	
IPOBRKA:	Критерий смены кадра: Рампа торможения		
	Активность:	модально	
ADISPOSA:	Окно допуска для критерия окончания движения		
	Активность:	модально	
<ось>:	Имя оси канала (X, Y,)		
<момент времени>:	Момент времени смены кадра, относительно ramпы торможения в %:		
	• 100% = начало ramпы торможения • 0% = конец ramпы торможения, однозначно с IPOENDA		
	Тип:	REAL	
<режим>:	Отношение окна допуска		
	Диапазон значений:	0	Окно допуска не активно
		1	Окном допуска относительно заданной позиции
		2	Окном допуска относительно фактической позиции
	Тип:	INT	
<размер окна>:	Размер окна допуска		
	Тип:	REAL	

5.11 Программируемый критерий окончания движения (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Примеры

Пример 1: Критерий окончания движения: "Стоп интерполятора"

Программный код
; переместить позиционирующую ось X на 100, скорость 200 м/мин, ускорение 90%, ; критерий окончания движения: Стоп интерполятора N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X] ; синхронное действие: ; ВСЕГДА КОГДА: вход 1 установлен ; ТОГДА переместить позиционирующую ось X на 50, скорость 200 м/ мин, ускорение 140%, ; критерий окончания движения: Стоп интерполятора N120 EVERY \$A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140 IPOENDA[X]

Пример 2: Критерий смены кадра: "Рампа торможения"

Программный код	Комментарий
N40 POS[X]=100	; установка по умолчанию действует ; Движение позиционирования X на позицию 100
N20 IPOBRKA(X,100)	Критерий смены кадра: точный останов точный ; Критерий смены кадра: "Рампа торможения", 100% = начало ramпы торможения
N30 POS[X]=200	; Смена кадра выполняется, как только ось X начинает торможение
N40 POS[X]=250	; Ось X не продолжает торможение на позицию 200, а продолжает движение на позицию 250. Как только ось начинает торможение, происходит смена кадра.
N50 POS[X]=0	; Ось X тормозит и возвращается на позицию 0. Смена кадра выполняется на позиции 0 и при "Точном останове точном"
N60 X10 F100	; Ось X движется как траекторная ось на позицию 10

Дополнительная информация

Системная переменная для критерия окончания движения

Активный критерий окончания движения может быть считан через системную переменную \$AA_MOTEND.

Литература: /LIS2sl/ Справочник по параметрированию, том 2

Критерий смены кадра: "Рампа торможения" (IPOBRKA)

Если при активации критерия смены кадра "Рампа торможения" для опционального момента времени смены кадры, но начинает действовать для следующего движения позиционирования и записывается синхронно с главным ходом в установочные данные.

5.11 Программируемый критерий окончания движения (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Если значение для момента времени смены кадра не указано, то активируется актуальное значение установочных данных.

SD43600 \$SA_IPOBRAKE_BLOCK_EXCHANGE

При следующем программировании осевого критерия окончания движения (FINEA, COARSEA, IPOENDA) IPOBRKA для соответствующей оси деактивируется.

Дополнительный критерий смены кадра: "Окно допуска" (ADISPOSA)

С ADISPOSA в качестве дополнительного критерия смены кадра можно определить окно допуска вокруг конечной точки кадра (по выбору фактическая или заданная позиция). Тогда для смены кадра должны быть выполнены оба условия:

- Критерий смены кадра: "Рампа торможения"
- Критерий смены кадра: "Окно допуска"

Литература

Дополнительную информацию по критерию смены кадров позиционирующих осей можно найти в:

- Описание функций "Расширенные функции"; Позиционирующие оси (P2)
- Руководство по программированию "Основы"; глава "Регулирование подачи"

Трансформации координат (фреймы)

6.1 Трансформация координат через фрейм-переменные

Наряду с командами, описанными в Руководстве по программированию "Основы", глава "Трансформации координат (фреймы)", такими как, например, `ROT`, `AROT`, `SCALE` и т. д., систему координат детали (WCS) можно трансформировать и через фрейм-переменные `$P_...FR` (фреймы УД) и `$P_...FRAME` (активные фреймы).

На следующем рисунке показана схема структурирования фрейм-переменных:

- Фреймы УД
- Активные фреймы
- Активный общий фрейм: соединение всех активных фреймов
- Глобальные фреймы ЧПУ
- Специфические для канала фреймы

6.1 Трансформация координат через фрейм-переменные

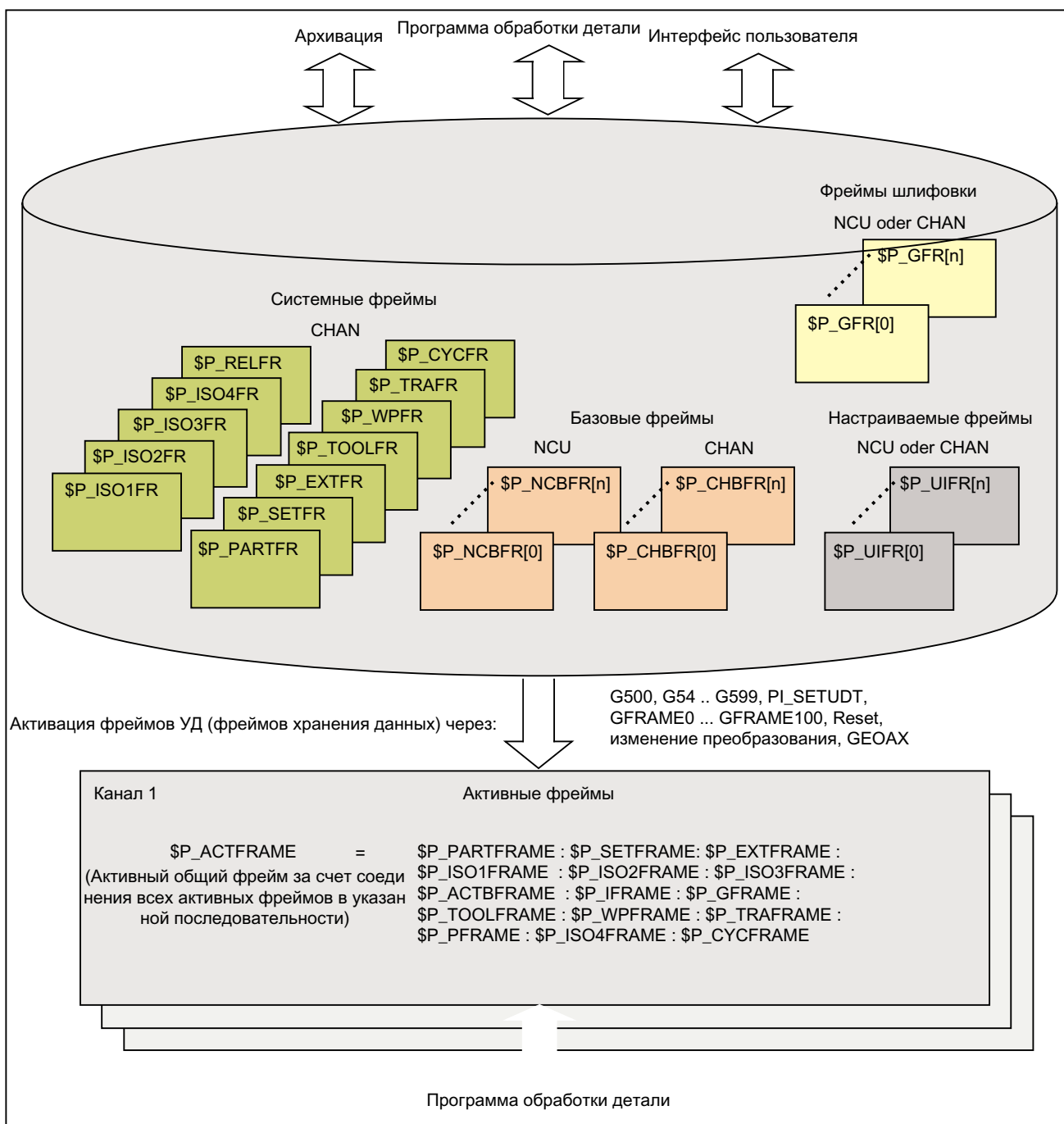


Рисунок 6-1 Обзор фрейм-переменных

6.1.1 Предопределенные фрейм-переменные (\$P_CHBFRAME, \$P_IFRAME, \$P_PFRAME, \$P_ACTFRAME)

Активно: Специфические для канала фреймы \$P_CHBFRAME[<n>] (\$P_BFRAME)

Примечание

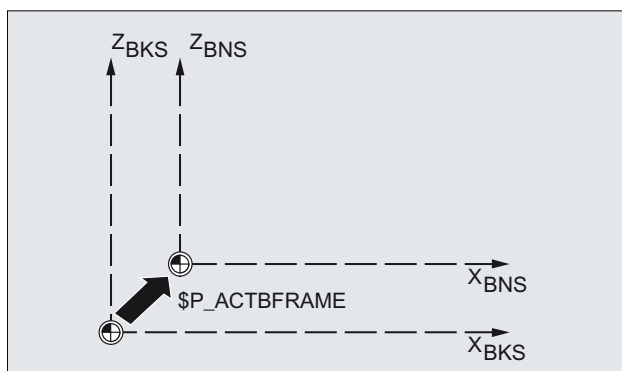
Актуальный базовый фрейм \$P_BFRAME и базовый фрейм УД \$P_UBFR сохраняются из соображений совместимости.

- $\$P_BFRAME \sqsubset \$P_CHBFRAME[0]$
- $\$P_UBFR \sqsubset \$P_CHBFR[0]$.

Фрейм-переменные \$P_CHBFRAME[<n>] определяют отношение между базовой кинематической системой (BKS) и базовой системой нулевой точки (BNS).

Если актуальный специфический для канала базовый фрейм \$P_CHBFRAME[<n>] должен активироваться в программе ЧПУ немедленно, имеются следующие возможности

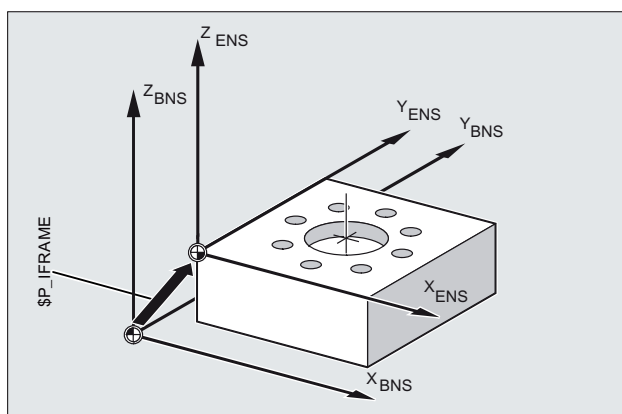
- Команды:
 - G500 (выключение всех настраиваемых фреймов, базовые фреймы остаются активными)
 - G54 ... G599 (настраиваемые смещения нулевой точки)
- Присваивание специфического для канала базового фрейма системы УД актуальному специфическому для канала базовому фрейму:
 $\$P_CHBFRAME[<n>] = \$P_CHBFR[<m>]$



Активно: специфический для канала фрейм \$P_IFRAME

Фрейм-переменная \$P_IFRAME, определяет отношение между базовой системой нулевой точки (BNS) и настраиваемой системой нулевой точки (ENS).

- \$P_IFRAME соответствует \$P_UIFR[\$P_IFRNUM]
- \$P_IFRAME содержит после программирования, к примеру, G54 определенные через G54 смещение, вращение, масштабирование и отражение.

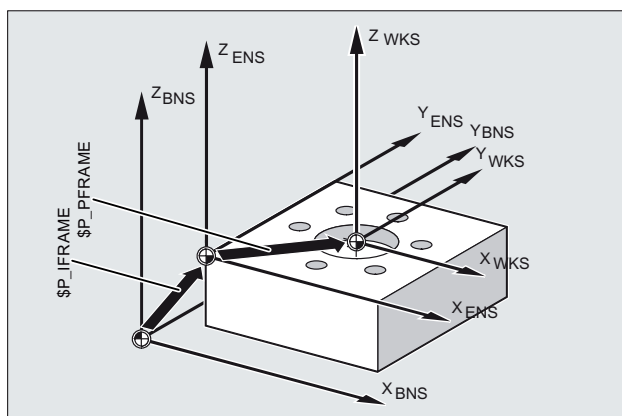


Активно: специфический для канала программируемый фрейм \$P_PFRAME

Фрейм-переменная \$P_PFRAME, определяет отношение между настраиваемой системой нулевой точки (ENS) и системой координат детали (WCS).

\$P_PFRAME содержит результирующий фрейм, полученный

- из программирования TRANS/ATRANS, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR или
- из присваивания CTRANS, CROT, CMIRROR, CSCALE программируемому FRAME



Активно: Общий фрейм \$P_ACTFRAME

Действующий в канале общий фрейм, получается из соединения всех действующих в канале фреймов.

```
$P_ACTFRAME =    $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :
                  $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :
                  $P_ACTBFRAME : $P_IFRAME : $P_GFRAME :
                  $P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME :
                  $P_PFRAME : $P_ISO4FRAME : $P_CYCFRAME
```

\$P_ACTFRAME описывает нулевую точку актуальной системы координат детали.

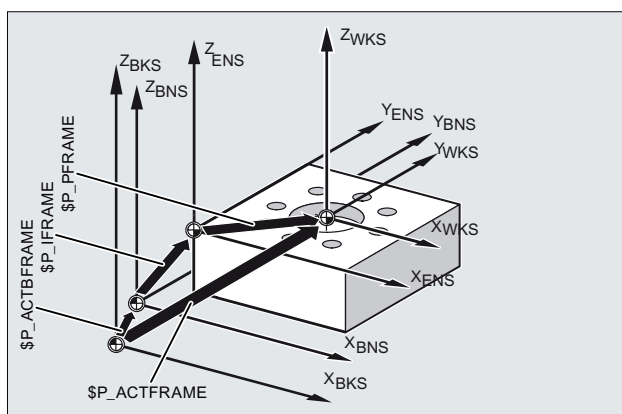
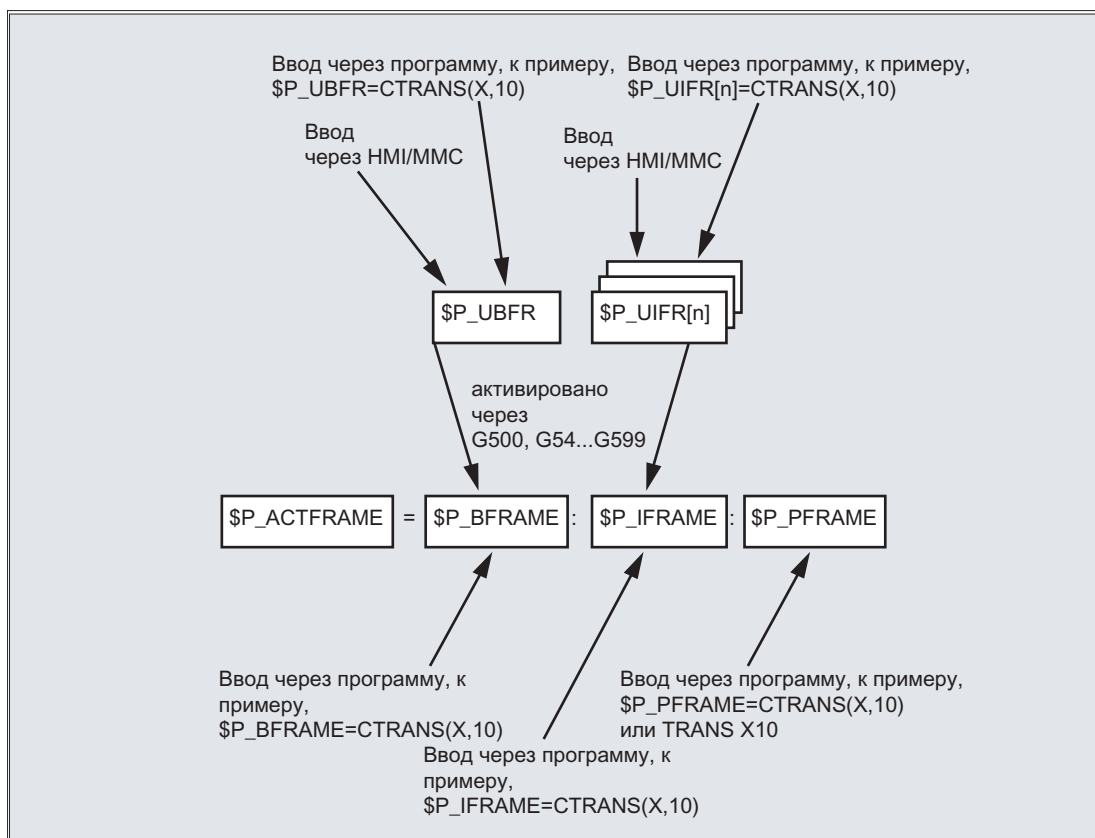


Рисунок 6-2 Фрейм-переменная \$P_ACTFRAME

Если один из следующих фреймов $\$P_BFRAME$ / $\$P_CHBFRAME[<n>]$, $\$P_IFRAME$ или $\$P_PFRAME$ изменяется, то актуальный общий фрейм $\$P_ACTFRAME$ пересчитывается заново.



Базовый фрейм и настраиваемый фрейм действуют после Reset, если MD 20110 RESET_MODE_MASK установлены следующим образом:

Бит0=1, Бит14=1 --> $\$P_UBFR$ (базовый фрейм) действует

Бит0=1, Бит5=1 --> $\$P_UIFR[\$P_UIFRNUM]$ (настр. фрейм) действует

Хранение данных: Специфические для канала базовые фреймы \$P_CHBFR[<n>]

Через фрейм-переменные \$P_CHBFR[<n>] базовые фреймы в системе УД считываются / записываются. Фрейм УД при записи активируется в канале не сразу. Активация записанного фрейма происходит при следующих условиях:

- Reset канала и MD20110 \$MC_RESET_MODE_MASK, Бит0 == 1 и Бит14 == 1
- Команда G500, G54 ... G57, G505 ... G599 (включение/выключение базовых фреймов с последующим пересчетом актуального общего фрейма)

Хранение данных: Специфические для канала настраиваемые фреймы \$P_UIFR[<n>]

Через фрейм-переменные \$P_UIFR[<n>] настраиваемые фреймы в системе УД считываются / записываются. Фрейм при записи активируется в канале не сразу. Записанный фрейм учитывается в канале при следующих условиях:

- Команда G500 (выключение всех настраиваемых фреймов или смещений нулевой точки)
- Команда G54 ... G57, G505 ... G599 (включение настраиваемого фрейма или смещения нулевой точки)

Активный настраиваемый фрейм	Фрейм УД	(соответствует команде)
\$P_IFRAME =	\$P_UIFR[0]	G500
	\$P_UIFR[1]	G54
	\$P_UIFR[2]	G55
	\$P_UIFR[3]	G56
	\$P_UIFR[4]	G57
	\$P_UIFR[5]	G505
	\$P_UIFR[6]	G506
	...	...
	\$P_UIFR[99]	G599

6.2 Присвоение значений фреймам

6.2.1 Присвоение прямых значений (значение оси, угол, масштаб)

В программе ЧПУ возможно прямое присваивание значений фреймам или фрейм-переменным.

Синтаксис

Синтаксис

```
$P_PFRAME = CTRANS (X, <значение смещения>, Y, <значение смещения>, Z, <значение смещения>, ...)
```

```
$P_PFRAME = ROT (X, <угол>, Y, <угол>, Z, <угол>, ...)
```

```
$P_UIFR[...] = CROT (X, <угол>, Y, <угол>, Z, <угол>, ...)
```

```
$P_PFRAME = CSCALE (X, <масштаб>, Y, <масштаб>, Z, <масштаб>, ...)
```

```
$P_PFRAME = CMIRROR (X, Y, Z)
```

Синтаксис для \$P_CHBFRAME [<n>] идентичен \$P_PFRAME.

Значение

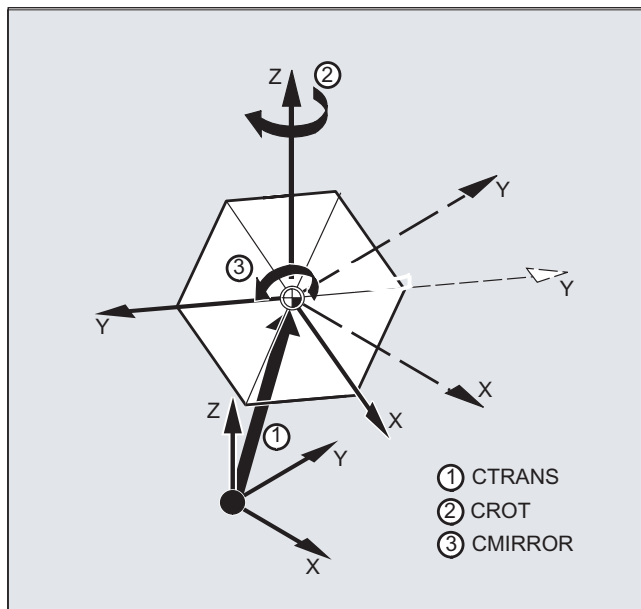
CTTRANS:	Смещение в указанных осях
CROT:	Вращение вокруг указанных осей
CSCALE:	Изменение масштаба в указанных осях
CMIRROR:	Реверсирование указанной оси
X, Y, Z :	Значение смещения в направлении указанной гео-оси
<значение смещения>:	Значение смещения
<угол>:	Угол, на который выполняется поворот
<масштаб>:	Указание масштаба

Примеры

Присваивание значений компонентам актуального программируемого фрейма

Присваивание значений компонентам смещения, поворота и отражения актуального программируемого фрейма:

```
$P_PFRAME = CTRANS (X, 10, Y, 20, Z, 5) : CROT (Z, 45) : CMIRROR (Y)
```



Запись компонентов поворота фрейма

Присваивание значений всем трем осям компонента поворота настраиваемого фрейма УД \$P_UIFR с помощью CROT :

```
$P_UIFR[5]=CROT (X, 0, Y, 0, Z, 0)
```

В качестве альтернативы этому прямое присваивание отдельных значений соответствующим осям компонента поворота фрейма УД:

```
$P_UIFR[5, Y, RT]=0  
$P_UIFR[5, X, RT]=0  
$P_UIFR[5, Z, RT]=0
```

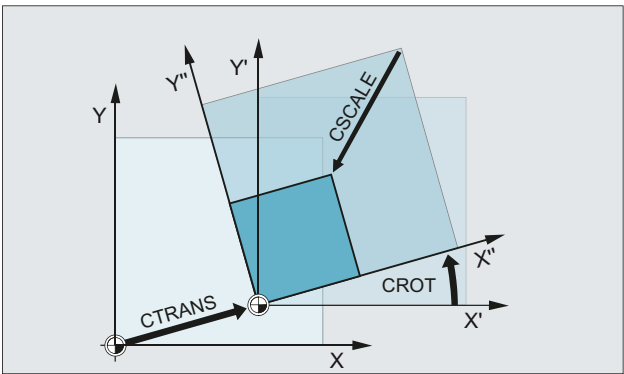
Описание

Несколько операций на одном фрейме можно соединить через связывающий оператор : друг с другом. В этом случае операции выполняются последовательно слева направо.

Пример

Связанные операции на \$P_PFRAME со смещением, поворотом и масштабированием:

```
$P_PFRAME = CTRANS(...) : CROT(...) : CSCALE...
```



6.2.2 Чтение и изменение компонентов фрейма (TR, FI, RT, SC, MI)

Существует возможность обращения к **отдельным** данным фрейма, к примеру, к определенному значению смещения или углу поворота. Эти значения могут изменяться или присваиваться другой переменной.

Синтаксис

<code>R10=\$P_UIFR[\$P_UIFNUM, X, RT]</code>	Угол поворота RT вокруг оси X из актуального действующего настраиваемого смещения нулевой точки \$P_UIFRNUM должен быть присвоен переменной R10.
<code>R12=\$P_UIFR[25, Z, TR]</code>	Значение смещения TR в Z из блока данных установленного фрейма Nr. 25 должно быть присвоено переменной R12.
<code>R15=\$P_PFRAME[Y, TR]</code>	Значение смещения TR в Y актуального программируемого фрейма должно быть присвоено переменной R15.
<code>\$P_PFRAME[X, TR] = 25</code>	Значение смещения TR в X актуального программируемого фрейма должно быть изменено. С этого момента действует X25.

Значение

\$P_UIFRNUM:	С помощью этой переменной автоматически создается связь с актуальным действительным настраиваемым смещением нулевой точки.
P_UIFR[n, ..., ...] :	Посредством указания номера фрейма n осуществляется обращение к настраиваемому фрейму Nr. n.
	Указание компонента, который должен быть считан или изменен:
TR:	TR смещение
FI:	FI смещение точное
RT:	RT вращение
SC:	SC Scale изменение масштаба

MI:	MI отражение
X, Y, Z:	Дополнительно (см. примеры) указывается соответствующая ось X, Y, Z.

Диапазон значений для вращения RT

Вращение вокруг 1-й гео-оси: -180° до +180°

Вращение вокруг 2-й гео-оси: -90° до +90°

Вращение вокруг 3-й гео-оси: -180° до +180°

Описание**Вызов фрейма**

Через указание системной переменной \$P_UIFRNUM можно напрямую обращаться к актуальному, установленному с помощью \$P_UIFR или G54, G55, ... смещению нулевой точки

(\$P_UIFRNUM содержит номер актуального установленного фрейма).

Все другие сохраненные настраиваемые фреймы \$P_UIFR вызываются через указание соответствующего номера \$P_UIFR[n].

Для предопределенных фрейм-переменных и определенных пользователем фреймов указывается имя, к примеру, \$P_IFRAME.

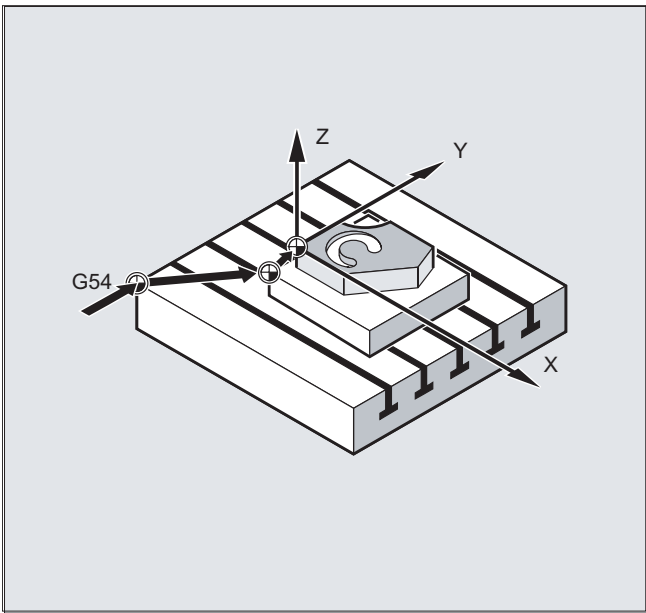
Вызов данных

В квадратных скобках стоят имя оси и компоненты фрейма значения, к которому выполняется обращение или которое должно быть изменено, к примеру, [X, RT] или [Z, MI].

6.2.3 Вычисление с фреймами

В программе ЧПУ возможно присвоение одного фрейма другому фрейму или соединение фреймов друг с другом.

Соединения фреймов подходят, к примеру, для описания нескольких деталей, расположенных на одной палете, которые должны быть обработаны в течение одного технологического процесса.



Для описания задач палет компоненты фрейма могут содержать, к примеру, только определенные частные значения, посредством соединения которых создаются различные нулевые точки детали.

Примеры

Присваивания

Программный код	Комментарий
DEF FRAME EINSTELLUNG_1	; Определение локальной фрейм-переменной
EINSTELLUNG_1 = CTRANS(X,10)	; Присваивание результата функции фрейм-переменной
\$P_PFRAME = EINSTELLUNG_1	; Присваивание фрейм-переменной актуальному фрейму
DEF FRAME EINSTELLUNG_4	; Определение локальной фрейм-переменной
EINSTELLUNG_4 = \$P_PFRAME	; Буферизация актуального фрейма во фрейм-переменной
...	
\$P_PFRAME = EINSTELLUNG_4	; Эхо-считывание актуального фрейма из фрейм-переменной

Соединения

Фреймы соединяются друг с другом в запрограммированной последовательности с помощью оператора : . Компоненты фрейма, например смещения, повороты и т. д., выполняются друг за другом аддитивно.

Программный код	Комментарий
\$P_IFRAME = \$P_UIFR[15] :	; Присваивание фрейма результата из соединения
\$P_UIFR[16]	; двух настраиваемых фреймов УД активному
	; настраиваемому общему фрейму.
	; Пример использования:
	; \$P_UIFR[15]: Смещение
	; \$P_UIFR[16]: Вращение

Программный код	Комментарий
\$P_UIFR[3] = \$P_UIFR[4] :	; Присваивание фрейма результата из соединения
\$P_UIFR[5]	; двух настраиваемых фреймов УД какому-либо
	; другому настраиваемому фрейму УД

6.2.4 Определение фрейм-переменных (DEF FRAME)

Наряду с предопределенными фрейм-переменными можно определять и собственные фрейм-переменные. Самостоятельно определяемые фрейм-переменные – это переменные пользователя типа FRAME. Имя фрейму можно давать произвольно в рамках правил работы с переменными пользователя.

С помощью функций CTRANS, CROT, CSCALE, CMIRROR самостоятельно определяемым фрейм-переменным можно присваивать значения.

Синтаксис

```
DEF FRAME <имя>
```

Значение

DEF FRAME:	Определить переменные пользователя типа FRAME.
<имя>:	Имя фрейм-переменной

Пример

Определение фрейм-переменной "PALETTE" и присваивание значений смещения и поворота:

Программный код	Комментарий
DEF FRAME PALETTE	; Определить фрейм-переменную PALETTE
PALETTE = CTRANS(...) : CROT(...)	; Присваивание фрейма результата соединения
	; смещения и поворота фрейм-переменной PALETTE

6.3 Грубое и точное смещение (CTRANS, CFINE)

Точное смещение

Точное смещение `CFINE ( . . . )` можно применять к следующим фреймам:

- Настраиваемые фреймы: `$P_UIFR` или `$P_IFRAME`
- Базовые фреймы: `$P_NCBFR[<n>]`, `$P_CHBFR[<n>]` или `$P_CHBFRAMES[<n>]` или `$P_ACTBFRAME`
- Программируемый фрейм: `$P_PFRAME`

Точное смещение фрейма программируется с командой `CFINE ( . . . )`.

Грубое смещение

Грубое смещение `CTRANS ( . . . )` можно применять ко всем фреймам.

Общее смещение

Общее смещение получается из сложения грубого и точного смещений.

Машинные данные

Разрешение точного смещения

Точное смещение разрешается с помощью машинных данных:

MD18600 \$MN_MM_FRAME_FINE_TRANS = 1

Синтаксис

Точное смещение

- Общий фрейм
 - `<фрейм> = CFINE (<K_1>, <значение>)`
 - `<фрейм> = CFINE (<K_1>, <значение>, <K_2>, <значение>)`
 - `<фрейм> = CFINE (<K_1>, <значение>, <K_2>, <значение>, <K_3>, <значение>)`
- Фрейм-компонент
 - `<фрейм>[<n>, <K_1>, FI] = <значение>`

Грубое смещение

- Общий фрейм
 - `<фрейм> = CTRANS (<K_1>, <значение>)`
 - `<фрейм> = CTRANS (<K_1>, <значение>, <K_2>, <значение>)`
 - `<фрейм> = CTRANS (<K_1>, <значение>, <K_2>, <значение>, <K_3>, <значение>)`
- Фрейм-компонент
 - `<фрейм>[<n>, <K_1>, TR] = <значение>`

Специально для программируемого фрейма \$P_PFRAME:

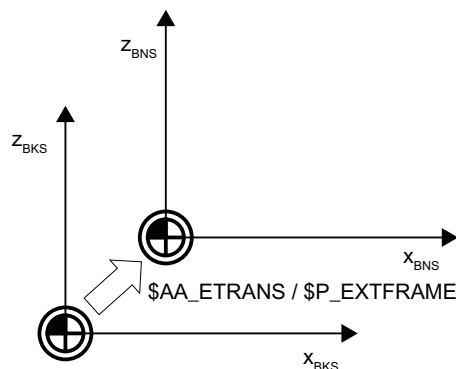
- TRANS <K_1> <значение>
- TRANS <K_1> <значение> <K_2> <значение>
- TRANS <K_1> <значение> <K_2> <значение> <K_3> <значение>

Значение

<фрейм>:	Фрейм, например настраиваемый фрейм системы УД \$P_UIFR[<n>]
CFine:	Точное смещение, аддитивное смещение.
CTrans:	Грубое смещение, абсолютное смещение
TRANS:	Только программируемый фрейм: Грубое смещение, абсолютное смещение
<K_n>:	Оси координат X, Y, Z
<значение>:	Значение смещения

6.4 Внешнее смещение нулевой точки (\$AA_ETRANS)

Внешнее смещение нулевой точки – это линейное смещение между базовой кинематической системой (BKS) и базовой системой нулевой точки (BNS).



Внешнее смещение нулевой точки с помощью \$AA_ETRANS действует двумя способами в зависимости от параметрирования машинных данных:

1. Системная переменная \$AA_ETRANS после активации интерфейсным сигналом ЧПУ/PLC действует непосредственно как значение смещения
2. Значение системной переменной \$AA_ETRANS после активации интерфейсным сигналом ЧПУ/PLC принимается в активные системные фреймы \$P:EXTFRAME и во фрейм УД \$P_EXTFR. Затем активный общий фрейм \$P_ACTFRAME пересчитывается заново.

Машинные данные

При использовании системной переменной \$AA_ETRANS следует различать два способа действий, которые выбираются через следующие машинные данные:

MD28082 \$MC_MM_SYSTEM_FRAME_MASK, бит1 = <значение>

<значение>	Объяснение
0	Функция: Прямая запись \$AA_ETRANS[<ось>] через PLC, HMI или программу ЧПУ. Разрешение на вывод смещения нулевой точки из \$AA_ETRANS[<ось>] в ближайшем кадре перемещения: DB31, ...DBX3.0
1	Функция: Активация активного системного фрейма \$P:EXTFRAME и фрейма УД \$P_EXTFR Разрешение на вывод смещения нулевой точки из \$AA_ETRANS[<ось>] через: DB31, ...DBX3.0. После этого в канале происходит: • Остановка всех движений перемещения в канале (кроме командных осей/осей PLC) • Остановка предварительной обработки с последующей реорганизацией (STOPRE) • Грубое смещение активного фрейма \$P_EXTFRAME[<ось>] = \$AA_ETRANS[<ось>] • Грубое смещение фрейма УД \$P_EXTFR[<ось>] = \$AA_ETRANS[<ось>] • Пересчет активного общего фрейма \$P_ACTFRAME • Вывод смещения в запрограммированных осях. • Продолжение прерванного движения перемещения или программы ЧПУ

Программирование

- Синтаксис
\$AA_ETRANS [<ось>] = <значение>
- Объяснение

\$AA_ETRANS:	Системная переменная для буферизации внешнего смещения нулевой точки
<ось>:	Ось канала
<значение>:	Значение смещения

Интерфейсный сигнал ЧПУ/PLC

DB31, ... DBX3.0 = 0 → 1 ⇒ \$P_EXTFRAME[<ось>] = \$P_EXTFR[<ось>] = \$AA_ETRANS[<ось>]

6.5 Установка фактического значения с потерей состояния реферирования (PRESETON)

Процедура `PRESETON()` устанавливает для одной или нескольких осей новое фактическое значение в системе координат станка (MCS). Это соответствует смещению нулевой точки MCS оси. При этом ось не перемещается.

С помощью `PRESETON` вызывается остановка предварительной обработки с синхронизацией. Фактическая позиция присваивается оси только в состоянии покоя.

Если ось при `PRESETON` не согласована с каналом, дальнейшие действия зависят от специфического для оси конфигурирования режима перехода оси:

MD30552 \$MA_AUTO_GET_TYPE

Состояние реферирования

При установке нового фактического значения в системе координат станка состояние реферирования оси станка сбрасывается:

DB31, ... DBX60.4 / .5 = 0 (Реферировать / синхронизировать измерительную систему 1 / 2)

Рекомендуется использовать `PRESETON` только для осей без обязательной референтной точки.

Для восстановления первоначальной системы координат станка необходимо заново реферировать измерительную систему оси станка, например, через реферирование из программы обработки детали (G74).



ОСТОРОЖНО

Потеря состояния реферирования

При установке нового фактического значения в системе координат станка с помощью `PRESETON` состояние реферирования оси станка сбрасывается на "не референцировано / синхронизировано".

Программирование

Синтаксис

`PRESETON (<ось_1>, <значение_1> [, <ось_2>, <значение_2>, ... <ось_8>, <значение_8>])`

Значение

PRESETON:	Установка фактического значения с потерей состояния реферирования	
	Остановка предварительной обработки:	да
	Автономно в кадре:	да
<ось_x>:	Имя оси станка	
	Тип:	AXIS
	Диапазон значений:	определенные в канале имени осей станка

<значение _x>:	Новое фактическое значение оси станка в системе координат станка (MCS) Ввод осуществляется в актуальной действующей системе единиц (дюймовой / метрической) Учитывается активное программирование диаметра (DIAMON)	
	Тип:	REAL

Литература

PRESETONS в программе ЧПУ

Подробное описание PRESETON в программах ЧПУ см.:

Описание функций "Основные функции"; глава "K2: Оси, системы координат, фреймы" > "Системы координат" > "Система координат станка (MCS)" > "Установка фактического значения с потерей состояния реферирования (PRESETON)"

PRESETONS в синхронных действиях

Подробное описание PRESETON в синхронных действиях см.:

Описание функций - Синхронные действия; глава: "Подробное описание" > "Действия в синхронных действиях" > "Установка фактического значения с потерей состояния реферирования (PRESETON)"

6.6 Установка фактического значения без потери состояния реферирования (PRESETONS)

Процедура `PRESETONS ( )` устанавливает для одной или нескольких осей новое фактическое значение в системе координат станка (MCS). Это соответствует смещению нулевой точки MCS оси. При этом ось не перемещается.

С помощью `PRESETONS` вызывается остановка предварительной обработки с синхронизацией. Фактическая позиция присваивается оси только в состоянии покоя.

Если ось при `PRESETONS` не согласована с каналом, дальнейшие действия зависят от специфического для оси конфигурирования режима перехода оси:

MD30552 \$MA_AUTO_GET_TYPE

Состояние реферирования

При установке нового фактического значения в системе координат станка (MKS) с помощью `PRESETONS` состояние реферирования оси станка **не** изменяется.

Условия

- **Тип датчика**

`PRESETONS` можно использовать только при следующих типах датчика активной измерительной системы:

- MD30240 \$MA_ENC_TYPE[<измерительная система>] = 0 (Симулированный датчик)
- MD30240 \$MA_ENC_TYPE[<измерительная система>] = 1 (Датчик необработанного сигнала)

- **Режим реферирования**

`PRESETONS` можно использовать только при следующих режимах реферирования активной измерительной системы:

- MD34200 \$MA_ENC_REFP_MODE[<измерительная система>] = 0 (Реферирование невозможно)
- MD34200 \$MA_ENC_REFP_MODE[<измерительная система>] = 1 (Реферирование инкрементных, ротационных или линейных измерительных систем: нулевой импульс в канале датчика)

Программирование

Синтаксис

`PRESETONS ( <ось_1>, <значение_1> [, <ось_2>, <значение_2>, ... <ось_8>, <значение_8>])`

Значение

PRESETONS:	Установка фактического значения без потери состояния реферирования	
	Остановка предварительной обработки:	да
	Автономно в кадре:	да

<ось _x>:	Имя оси станка	
	Тип:	AXIS
	Диапазон значений:	определенные в канале имени осей станка
<значение _x>:	Новое актуальное фактическое значение оси станка в системе координат станка (MCS) Ввод осуществляется в активной системе единиц (дюймовой / метрической) Учитывается активное программирование диаметра (DIAMON)	
	Тип:	REAL

Литература

PRESETONS в программе ЧПУ

Подробное описание PRESETONS в программах ЧПУ см.:

Описание функций "Основные функции"; глава "K2: Оси, системы координат, фреймы" > "Системы координат" > "Система координат станка (MCS)" > "Установка фактического значения без потери состояния реферирования (PRESETONS)"

PRESETONS в синхронных действиях

Подробное описание PRESETONS в синхронных действиях см.:

Описание функций - Синхронные действия; глава: "Подробное описание" > "Действия в синхронных действиях" > "Установка фактического значения без потери состояния реферирования (PRESETONS)"

6.7 Вычисление фрейма из 3 точек измерения в пространстве (MEAFRAME)

Функция MEAFRAME используется для поддержки измерительных циклов. Она вычисляет фрейм из трех идеальных и соответствующих измеренных точек.

При позиционировании детали для обработки ее позиция относительно декартовой системы координат станка касательно ее идеальной позиции по умолчанию как смещена, так и повернута. Для точной обработки или измерения необходима либо дорогостоящая физическая юстировка, либо изменение движений в программе обработки детали.

Фрейм может быть определен через сканирование трех точек в пространстве, идеальные позиции которых известны. Сканирование выполняется посредством контактного или оптического сенсора, который касается точно зафиксированных на несущей пластине отверстий или измерительных сфер.

Синтаксис

MEAFRAME (<идеальные точки>, <точки измерения>, <качество>)

Значение

MEAFRAME:	Вызов функции		
<идеальные точки>:	2-мерное REAL-поле, содержащее три координаты идеальных точек		
<точки измерения>:	2-мерное REAL-поле, содержащее три координаты измеренных точек		
<качество>:	Переменная, возвращающая информацию о качестве вычисления ФРЕЙМА		
	Тип:	VAR REAL	
	Значение:	-1	Идеальные точки располагаются практически на одной прямой: Не удалось вычислить фрейм. Возвращенная ФРЕЙМ-переменная содержит нейтральный фрейм.
		-2	Точки измерения располагаются практически на одной прямой: Не удалось вычислить фрейм. Возвращенная ФРЕЙМ-переменная содержит нейтральный фрейм.
		-4	Вычисление матрицы вращения не удалось по иной причине.
		≥ 0.0	Сумма искажений (расстояния между точками), необходимая для перевода измеренного треугольника в конгруэнтный идеальному треугольнику.

Примечание
Качество измерения
Для того, чтобы измеренные координаты могли бы быть согласованы с идеальными координатами с комбинированным вращением/смещением, образованный точками измерения треугольник должен быть конгруэнтен идеальному треугольнику. Это осуществляется посредством алгоритма компенсации, минимизирующего сумму квадратов погрешностей, преобразующих измеренный треугольник в идеальный.
Эффективное необходимое искажение точек измерения может служить индикатором качества измерения и поэтому выводится как дополнительная переменная MEAFRAME.
Примечание
Созданный MEAFRAME фрейм через функцию ADDFRAME может быть преобразован в другой фрейм цепочки фреймов (см. пример "Соединение с ADDFRAME").

Примеры

Пример 1:

Программа обработки детали 1:

Программный код	Комментарий
... DEF FRAME CORR_FRAME	
DEF REAL IDEAL_POINT[3,3]= SET(10.0,0.0,0.0,0.0,10.0,0.0,0.0,0.0,10.0)	
DEF REAL MEAS_POINT[3,3]= SET(10.1,0.2,-0.2,-0.2,10.2,0.1,-0.2,0.2,9.8)	; Для теста.
DEF REAL FIT_QUALITY=0 DEF REAL ROT_FRAME_LIMIT=5	; Макс. разрешенный поворот позиции детали на 5 градусов.
DEF REAL FIT_QUALITY_LIMIT=3	; Разрешено макс. смещение в 3 мм между идеальным и измеренным треугольником.
DEF REAL SHOW_MCS_POS1[3] DEF REAL SHOW_MCS_POS2[3] DEF REAL SHOW_MCS_POS3[3]	

6.7 Вычисление фрейма из 3 точек измерения в пространстве (MEAFRAME)

Программный код	Комментарий
N100 G01 G90 F5000	
N110 X0 Y0 Z0	
N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS_POINT,FIT_QUALITY)	
N230 IF FIT_QUALITY < 0	
SETAL(65000)	
GOTOF NO_FRAME	
ENDIF	
N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT	
SETAL(65010)	
GOTOF NO_FRAME	
ENDIF	
N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT	; Ограничение 1-го угла RPY.
SETAL(65020)	
GOTOF NO_FRAME	
ENDIF	
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT	; Ограничение 2-го угла RPY.
SETAL(65021)	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT	; Ограничение 3-го угла RPY.
SETAL(65022)	
GOTOF NO_FRAME	
ENDIF	
N300 \$P_IFRAME=CORR_FRAME	; Активировать фрейм выборки с устанавливаемым фреймом. ; Проверить фрейм через позиционирование гео-осей на идеальные точки.
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=\$AA_IM[X]	
N420 SHOW_MCS_POS1[1]=\$AA_IM[Y]	
N430 SHOW_MCS_POS1[2]=\$AA_IM[Z]	
N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1] Z=IDEAL_POINT[1,2]	
N510 SHOW_MCS_POS2[0]=\$AA_IM[X]	
N520 SHOW_MCS_POS2[1]=\$AA_IM[Y]	
N530 SHOW_MCS_POS2[2]=\$AA_IM[Z]	
N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1] Z=IDEAL_POINT[2,2]	
N610 SHOW_MCS_POS3[0]=\$AA_IM[X]	
N620 SHOW_MCS_POS3[1]=\$AA_IM[Y]	
N630 SHOW_MCS_POS3[2]=\$AA_IM[Z]	
N700 G500	; Деактивировать устанавливаемый фрейм, так как предустановка с нулевым фреймом (значение не внесено).

Программный код	Комментарий
No_FRAME	; Деактивировать устанавливаемый фрейм, так как предустановка с нулевым фреймом (значение не внесено).
M0	
M30	

Пример 2: Соединение фреймов

Соединение MEAFRAME для коррекций

Функция MEAFRAME выводит фрейм коррекции. Если этот фрейм коррекции связывается с настраиваемым фреймом \$P_UIFR[1], который был активен при вызове функции (например, G54), то получается настраиваемый фрейм для дальнейших пересчетов для перемещения или обработки.

Соединение с ADDFRAME

Если этот фрейм коррекции должен действовать в цепочке фреймов на другом месте или если перед настраиваемым фреймом есть другие активные фреймы, то функция ADDFRAME для включения в цепочку может использоваться в одном из базовых фреймов канала или системном фрейме.

При этом во фреймах не должны быть активны:

- Отражение с MIRROR
- Масштабирование с SCALE

Входные параметры для заданных и фактических значений стоят в координатах детали. В основной системе СЧПУ эти координаты всегда должны указываться метрическими или дюймовыми (G71/G70) и как относящийся к радиусу (DIAMOF) размер.

Литература:

Дополнительную информацию по ADDFRAME см.:

Описание функций - Основные функции; K2: Оси, системы координат, фреймы

6.8 Глобальные фреймы NCU

Глобальные фреймы ЧПУ существуют для каждого ЧПУ только в одном экземпляре для всех каналов. Глобальные фреймы ЧПУ могут записываться и считываться из всех каналов. Активация глобальных фреймов ЧПУ осуществляется в соответствующем канале.

Посредством глобальных фреймов возможно масштабирование и отражение **осей канала и осей станка** со смещениями.

Геометрические связи и цепочки фреймов

Для глобальных фреймов не существует геометрической связи между осями. Поэтому выполнение вращений и программирование идентификаторов гео-осей невозможно.

- К глобальным фреймам не могут применяться вращения. Программирование вращения отклоняется с ошибкой: "18310 канал %1 кадр %2 фрейм: вращение недопустимо".
- Соединение глобальных фреймов и спец. для канала фреймов возможно. Результирующий фрейм содержит все компоненты фрейма вкл. вращения для всех осей. Присваивание фрейма с компонентами вращения глобальному фрейму отклоняется с ошибкой "Фрейм: вращение недопустимо".

Глобальные фреймы ЧПУ

Глобальные базовые фреймы УЧПУ \$P_NCBFR[n]

Может быть сконфигурировано до 8 глобальных базовых фреймов ЧПУ:

Одновременно возможно наличие спец. для канала базовых фреймов.

Глобальные фреймы могут записываться и считываться из всех каналов ЧПУ. При записи глобальных фреймов пользователь должен обеспечить координацию каналов. Это может быть реализовано, к примеру, с помощью меток ожидания (WAITMC).

Изготовитель станка

Число глобальных базовых фреймов проектируется через машинные данные.

Литература:

Описание функций - Основные функции; Оси, системы координат, фреймы (K2)

Глобальные настраиваемые фреймы ЧПУ \$P_UIFR[n]

Все настраиваемые фреймы G500, G54...G599 могут проектироваться глобально для ЧПУ или спец. для канала.

Изготовитель станка

Все настраиваемые фреймы с помощью машинных данных MD18601 \$MN_MM_NUM_GLOBAL_USER_FRAMES могут быть перепроектированы в глобальные фреймы.

В качестве идентификаторов осей для программных команд фреймов могут использоваться идентификаторы осей канала и идентификаторы осей станка. Программирование идентификаторов гео-осей отклоняется с ошибкой.

6.8.1 Специфические для канала фреймы (\$P_CHBFR, \$P_UBFR)

Настраиваемые фреймы или базовые фреймы могут записываться и считываться через программу обработки детали и через BTSS с панели оператора и с PLC.

Точное смещение возможно и для глобальных фреймов. Подавление глобальных фреймов, как и спец. для канала фреймов, осуществляется через G53, G153, SUPA и G500.

Изготовитель станка

Через машинные данные MD28081 \$MC_MM_NUM_BASE_FRAMES можно спроектировать число базовых фреймов в канале. Стандартная конфигурация такова, что существует мин. один базовый фрейм на канал. Возможно макс. 8 базовых фреймов на канал. В дополнение к 8 базовым фреймам в канале может существовать еще 8 глобальных базовых фреймов ЧПУ.

Специфические для канала фреймы

\$P_CHBFR[n]

Через системную переменную \$P_CHBFR[n] возможно чтение и запись базовых фреймов. При записи базового фрейма связанный общий базовый фрейм не активируется, а активация происходит только при выполнении оператора G500, G54 ... G599. Переменная служит в первую очередь памятью для процессов записи в базовый фрейм с HMI или PLC. Эти фрейм-переменные сохраняются через архивацию данных.

Первый базовый фрейм в канале

Запись в предопределенную переменную \$P_UBFR активирует базовый фрейм с индексом поля 0 не сразу же, а активация выполняется только при выполнении оператора G500, G54 ... G599. Запись и чтение переменной возможно и в программе.

\$P_UBFR

\$P_UBFR идентична \$P_CHBFR[0]. Стандартно всегда имеется один базовый фрейм в канале, поэтому системная переменная является совместимой с более старыми версиями. Если спец. для канала базовый фрейм отсутствует, то при записи или чтении выводится аварийное сообщение "Фрейм: недопустимый оператор".

6.8.2 Действующие в канале фреймы

Действующие в канале фреймы вводятся из программы обработки детали через соответствующие системные переменные этих фреймов. К ним относятся и системные фреймы. Через эти системные переменные возможно чтение и запись актуального системного фрейма в программе обработки детали.

Актуальные действующие в канале фреймы**Обзор****Актуальные системные фреймы**

\$P_PARTFRAME

\$P_SETFRAME

\$P_EXTFRAME

\$P_NCBFRAME[n]

\$P_CHBFRAME[n]

\$P_BFRAME

\$P_ACTBFRAME

\$P_CHBFRMASK и \$P_NCBFRMASK

\$P_IFFRAME

Актуальные системные фреймы

\$P_TOOLFRAME

\$P_WPFRAME

\$P_TRAFRAME

\$P_PFRAME

Актуальный системный фрейм

\$P_CYCFRAME

P_ACTFRAME

Цепочка фреймов

для:

TCARR и PAROT

Установка фактического значения и ка-
сание

Внешнее смещение нулевой точки

Актуальные глобальные базовые
фреймы ЧПУ

Актуальные базовые фреймы канала

Актуальный 1-й базовый фрейм в кана-
ле

Общий базовый фрейм

Общий базовый фрейм

Актуальный настраиваемый фрейм

для:

TOROT и TOFRAME

Исходные точки детали

Трансформации

Актуальный программируемый фрейм

для:

Циклы

Актуальный общий фрейм

Актуальный фрейм состоит из общего
базового фрейма**\$P_NCBFRAME[n] Актуальные глобальные базовые фреймы ЧПУ**

Через системную переменную \$P_NCBFRAME[n] возможно чтение и запись элементов поля актуальных глобальных базовых фреймов. Результирующий общий базовый фрейм через процесс записи учитывается в канале.

Измененный фрейм активируется только в том канале, в котором он был запрограммирован. Если необходимо изменить фрейм для всех каналов ЧПУ, то необходима одновременная запись в \$P_NCBFR[n] и \$P_NCBFRAME[n]. После проче-
с каналов еще должны активировать фрейм, к примеру, с G54. При записи базового
фрейма общий базовый фрейм вычисляется заново.

\$P_CHBFRAME[n] Актуальные базовые фреймы канала

Через системную переменную \$P_CHBFRAME[n] возможно чтение и запись элементов поля базовых фреймов канала. Результирующий общий базовый фрейм через процесс записи учитывается в канале. При записи базового фрейма общий базовый фрейм вычисляется заново.

\$P_BFRAME Актуальный 1-й базовый фрейм в канале

Через предопределенную фрейм-переменную $\$P_BFRAME$ актуальный базовый фрейм с индексом поля 0, действующий в канале, может быть считан и записан в программе обработки детали. Записанный базовый фрейм сразу же учитывается.

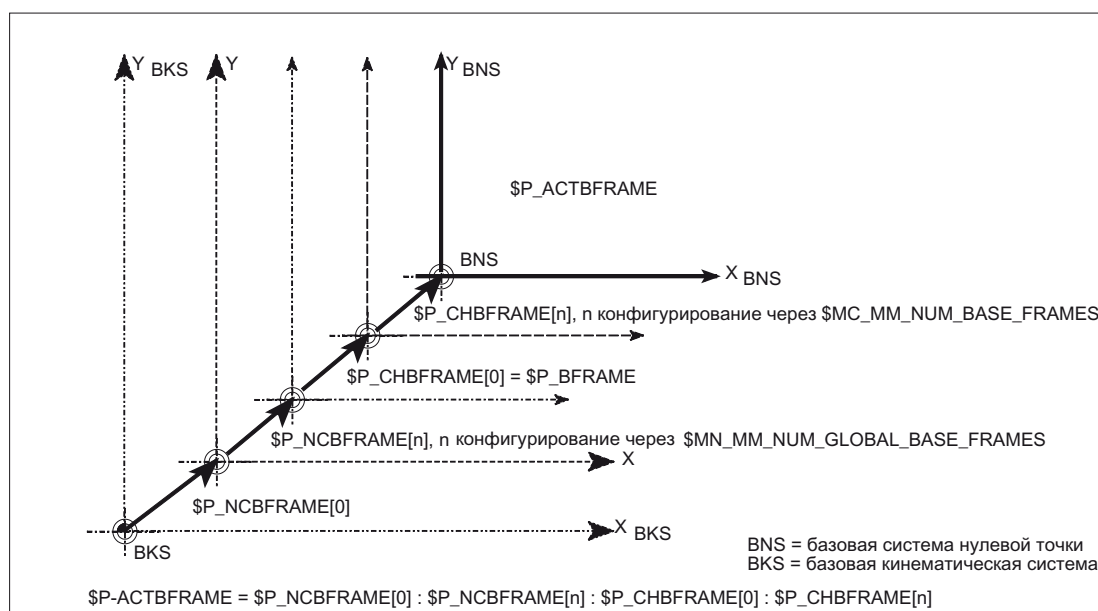
$\$P_BFRAME$ идентична $\$P_CHBFRAME[0]$. Системная переменная стандартно всегда имеет действительное значение. Если спец. для канала базовый фрейм отсутствует, то при записи или чтении выводится аварийное сообщение "Фрейм: недопустимый оператор".

\$P_ACTBFRAME Общий базовый фрейм

Переменная $\$P_ACTFRAME$ определяет связанный общий базовый фрейм. Переменная только для чтения.

$\$P_ACTFRAME$ соответствует:

$\$P_NCBFRAME[0] : \dots : \$P_NCBFRAME[n] : \$P_CHBFRAME[0] : \dots : \$P_CHBFRAME[n]$.

**\$P_CHBFRMASK** и **\$P_NCBFRMASK** Общий базовый фрейм

Через системную переменную $\$P_CHBFRMASK$ и $\$P_NCBFRMASK$ пользователь может выбрать, какие базовые фреймы он хотел бы включить в вычисление "общего" базового фрейма. Переменные могут программироваться только в программе и считываться через BTSS. Значение переменных интерпретируется как битовая маска и указывает, какой элемент поля базового фрейма из $\$P_ACTFRAME$ включается в вычисление.

С $\$P_CHBFRMASK$ можно установить, какие спец. для канала базовые фреймы, а с $\$P_NCBFRMASK$, какие базовые фреймы ЧПУ будут учтены.

При программировании переменных общий базовый фрейм и общий фрейм рассчитываются заново. После Reset и в первичной установке $\$P_CHBFRMASK$ и $\$P_NCBFRMASK$ имеют следующее значение:

```
$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK
```

```
$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK
```

Пример:

```
$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0] : $P_NCBFRAME[7]
```

```
$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0] : $P_CHBFRAME[4]
```

\$P_IFRAME Актуальный настраиваемый фрейм

Через предопределенную фрейм-переменную \$P_IFRAME актуальный настраиваемый фрейм, действующий в канале, может быть считан и записан в программе обработки детали. Записанный настраиваемый фрейм сразу же учитывается.

Для настраиваемых глобальных фреймов ЧПУ измененный фрейм действует только в канале, в котором фрейм был запрограммирован. Если фрейм должен быть изменен для всех каналов ЧПУ, то потребуется одновременная запись в \$P_UIFR[n] и \$P_IFRAME. После проче каналы еще должны активировать соответствующий фрейм, к примеру, с G54.

\$P_PFRAME Актуальный программируемый фрейм

\$P_PFRAME это программируемый фрейм, получаемый из программирования TRANS/ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR или из присваивания CTRANS, CROT, CMIRROR, CSCALE программируемому ФРЕЙМУ.

Актуальная, программируемая фрейм-переменная, устанавливающая отношение между настраиваемой системой нулевой точки (ENS) и системой координат детали (WCS).

P_ACTFRAME Актуальный общий фрейм

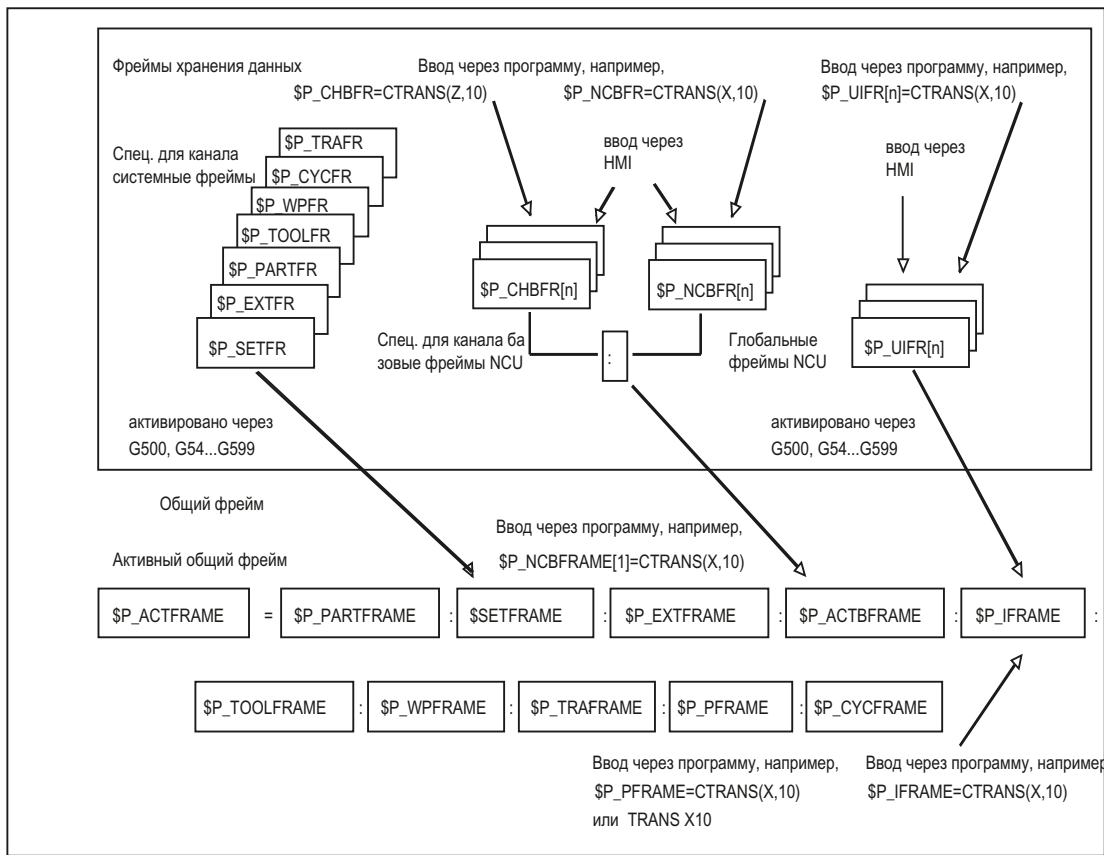
Актуальный результирующий общий фрейм \$P_ACTFRAME получается как соединение всех базовых фреймов, актуального настраиваемого фрейма и программируемого фрейма. Актуальный фрейм актуализируется всегда, когда изменяется один из его компонентов.

\$P_ACTFRAME соответствует:

```
$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :
```

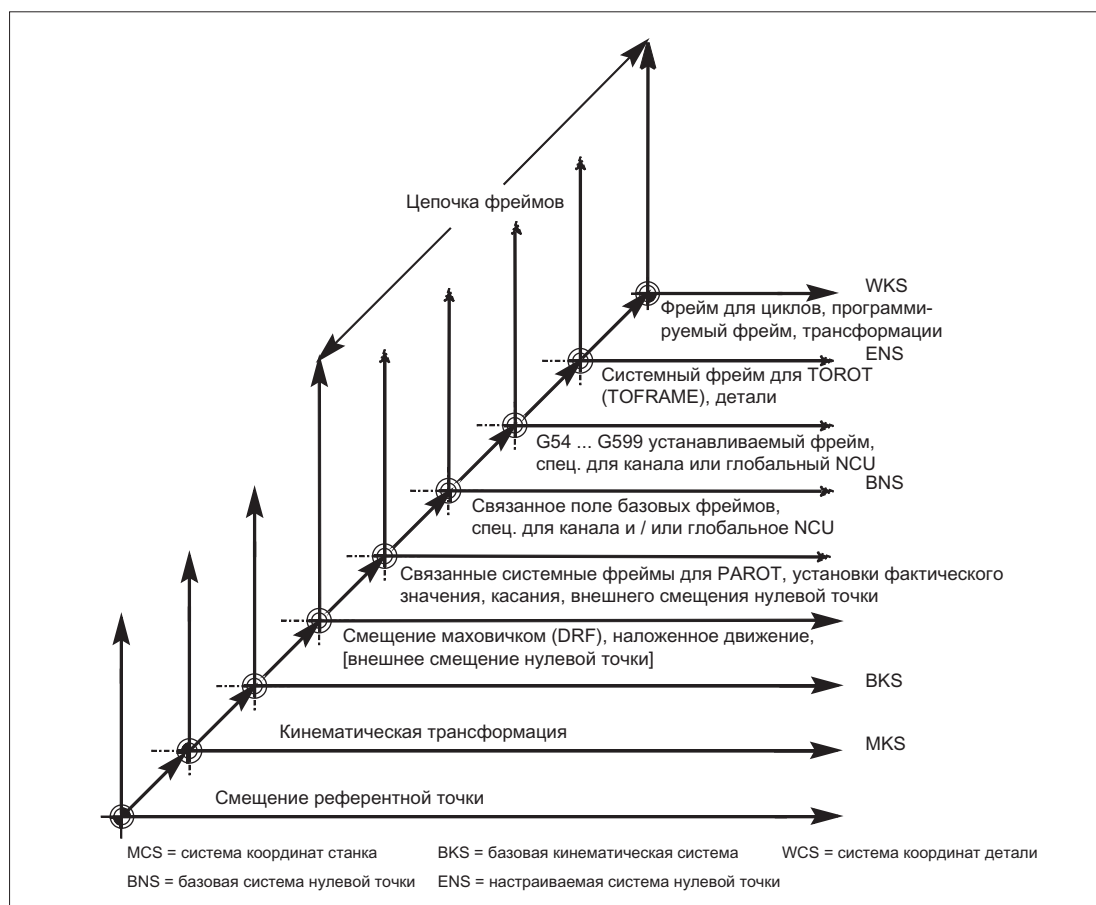
```
$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME
```

6.8 Глобальные фреймы NCU



Цепочка фреймов

Актуальный фрейм состоит из общего базового фрейма, настраиваемого фрейма, системного фрейма и программируемого фрейма согласно указанному выше актуальному общему фрейму.



Трансформации

7.1 Общее программирование типов трансформаций

Общая функция

Для согласования СЧПУ с различными кинематиками станков на выбор имеется несколько типов трансформаций с подходящими параметрами для программирования. Через эти параметры для выбранной трансформации можно соответственно согласовать как ориентацию инструмента в пространстве, так и движения ориентации круговых осей.

При 3-, 4- и 5-осевой трансформации запрограммированные данные позиций всегда относятся к острию инструмента, который отслеживается ортогонального к находящейся в пространстве обрабатываемой поверхности. Декартовы координаты пересчитываются из базовой кинематической системы в систему координат станка и относятся к гео-осям. Они описывают рабочую точку. Виртуальные круговые оси описывают ориентации инструмента в пространстве и программируются с TRAORI.

При кинематической трансформации позиции могут программироваться в декартовой системе координат. СЧПУ преобразует запрограммированные с TRANSMIT, TRACYL и TRAANG движения перемещения декартовой системы координат в движения перемещения реальных осей станка.

Программирование

3-, 4- и 5-осевые трансформации TRAORI

Согласованная трансформация ориентации активируется с помощью команды TRAORI и трех возможных параметров для номера трансформации, вектора ориентации и смещения круговой оси.

TRAORI (номер трансформации, вектор ориентации, смещение круговой оси)

Кинематические трансформации

К кинематическим трансформациям относятся согласованные трансформации

TRANSMIT (номер трансформации)

TRACYL (рабочий диаметр, номер трансформации)

TRAANG (угол наклонной оси, номер трансформации)

Отключить активную трансформацию

С помощью TRAFOOF активная в данный момент трансформация может быть отключена.

Трансформация ориентации

3-, 4- и 5-осевые трансформации TRAORI

Для оптимальной обработки оформленных в пространстве поверхностей в рабочей зоне станка, станку, кроме трех линейных осей X, Y и Z необходимы дополнительные оси. Дополнительные оси описывают ориентацию в пространстве и далее будут называться "оси ориентации". В качестве осей вращения они доступны для четырех типов станков с различной кинематикой.

1. Двухосевая качающаяся головка, к примеру, карданная инструментальная головка с круговой осью параллельно линейной оси при зафиксированном инструментальном столе.
2. Двухосевой поворотный стол, к примеру, зафиксированная качающаяся головка с вращающимся вокруг двух осей инструментальным столом.
3. Одноосевая качающаяся головка и одноосевой поворотный стол, к примеру, вращающаяся качающаяся головка с повернутым инструментом при вращающемся вокруг одной оси инструментальном столе.
4. Двухосевая качающаяся головка и одноосевой поворотный стол, к примеру, при вращающемся вокруг одной оси инструментальном столе и вращающейся качающейся головке с вращающимся вокруг самого себя инструментом.

3- и 4-осевые трансформации это особые формы 5-осевой трансформации, которые программируются аналогично 5-осевым трансформациям.

Объем функций "**Базовой 3-/4-/5-/6-осевая трансформации**" включает в себя расположенные под прямым углом круговые оси, а также трансформации для карданной фрезерной головки, она, как и любая другая трансформация ориентации может быть активирована и для этих четырех типов станков с TRAORI. При базовой 5/6-осевой трансформации ориентация инструмента имеет дополнительную третью степень свободы, при которой инструмент может вращаться вокруг собственной оси произвольно в пространстве в направлении инструмента.

Литература: /FB3/ Описание функций "Специальные функции"; 3- до 5-осевая трансформация (F2)

Независящая от кинематики первичная установка ориентации инструмента

ORIRESET

Если с TRAORI активна трансформация ориентации, то с помощью ORIRESET можно указать первичные установки макс. 3 осей ориентации с опционными параметрами A, B, C. Согласование последовательности запрограммированных параметров с круговыми осями выполняется согласно определенной через трансформацию последовательности осей ориентации. Следствием программирования ORIRESET(A, B, C) является линейное и синхронное движение осей ориентации от их актуальной позиции к указанной позиции первичной установки.

Кинематические трансформации

TRANSMIT и TRACYL

Для фрезерных обработок на токарных станках для согласованной трансформации можно запрограммировать либо

1. торцовую обработку в токарном зажиме с TRANSMIT, либо
2. обработку пазов произвольной формы на цилиндрических телах с TRACYL

TRAANG

Если необходима возможность, к примеру, для шлифовальной технологии, подачи оси подачи и под углом, то с помощью TRAANG для согласованной трансформации можно запрограммировать параметризуемый угол.

Движение "от точки к точке" в декартовой системе координат

К кинематической трансформации относится и "Движение от точки к точке в декартовой системе координат", при котором может быть запрограммировано до 8 положений шарнира STAT=. Позиции программируются в декартовой системе координат, при этом движение станка осуществляется в координатах станка.

Литература:

/FB2/ Описание функций "Расширенные функции"; Кинематическая трансформация (M1)

Последовательная связь трансформаций

Возможно последовательное включение двух трансформаций соответственно. Для связанной таким образом второй трансформации компоненты движения осей берутся из первой трансформации.

В качестве первой трансформации возможны:

- Трансформация ориентации TRAORI
- Полярная трансформация TRANSMIT
- Цилиндрическая трансформация TRACYL
- Трансформация наклонной оси TRAANG

Второй трансформацией должна быть "наклонная ось" TRAANG.

7.1.1 Движения ориентации при трансформациях

Движения перемещения и движения ориентации

Движения перемещения программируемых ориентаций зависят, в первую очередь, от типа станка. При 3-, 4- и 5-осевой трансформации с TRAORI круговые оси или поворотные линейные оси описывают движения ориентации инструмента.

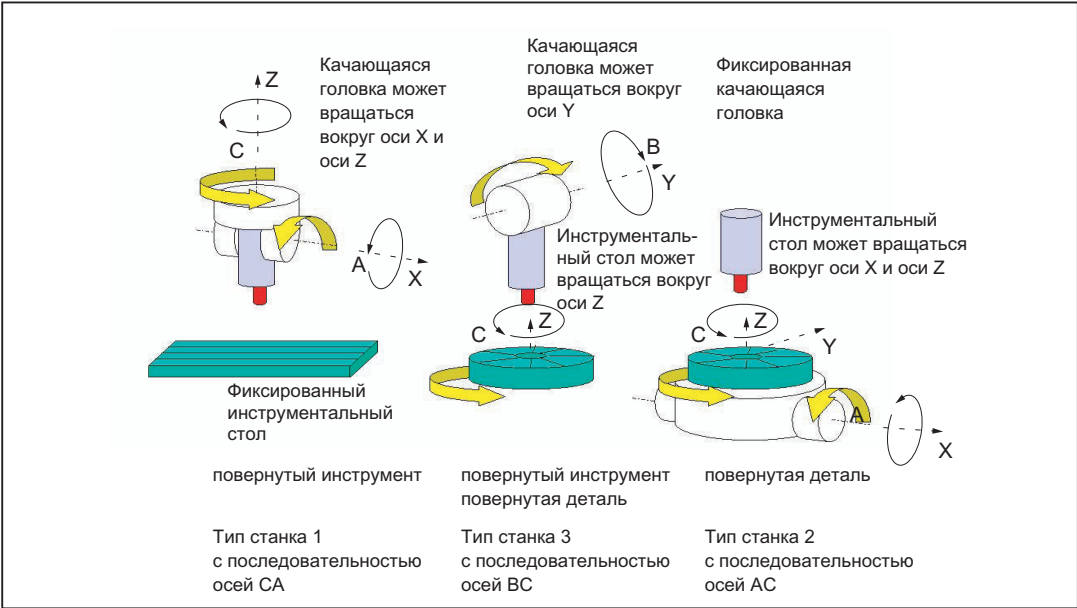
Изменения позиций участвующих в трансформации ориентации круговых осей приводят к движениям компенсации прочих осей станка. При этом позиция острия инструмента остается неизменной.

Движения ориентации инструмента могут программироваться через идентификаторы круговых осей A..., B..., C... виртуальных осей, в зависимости от использования, либо через указание эйлера угла или угла RPY или векторов направления или векторов нормалей плоскостей, нормированных векторов для оси вращения конуса, либо для промежуточной ориентации на боковой поверхности конуса.

При кинематической трансформации с TRANSMIT, TRACYL и TRAANG СЧПУ преобразует запрограммированные движения перемещения декартовой системы координат в движения перемещения реальных осей станка.

Кинематика станка при 3-, 4- и 5-осевой трансформации TRAORI

Возможно вращение либо инструмента, либо инструментального стола с помощью макс. двух круговых осей. Комбинация одноосевой качающейся головки и поворотного стола также возможна.



Тип станка	Программирование ориентации
3-осевая трансформация, типы станка 1 и 2	Программирование ориентации инструмента только в плоскости, расположенной вертикально к круговой оси. Существует две поступательные оси (линейные оси) и одна вращательная ось (круговая ось).
4-осевая трансформация, типы станка 1 и 2	Программирование ориентации инструмента только в плоскости, расположенной вертикально к круговой оси. Существует три поступательные оси (линейные оси) и одна вращательная ось (круговая ось).
5-осевая трансформация, типы станков 3 Одноосевая качающаяся головка и одноосевой поворотный стол	Программирование трансформации ориентации. Кинематика с тремя линейными осями и двумя ортогональными круговыми осями. Круговые оси расположены параллельно двум из трех линейных осей. Первая круговая ось приводится в движение двумя декартовыми линейными осями. Она вращает третью линейную ось с инструментом. Вторая круговая ось вращает деталь.

Базовые 5/6-осевые трансформации

Тип станка	Программирование трансформации ориентации
Базовая 5-/6-осевая трансформация, тип станка 4 Двухосевая качающаяся головка с вращающимся вокруг самого себя инструментом и одноосевой поворотный стол	Программирование трансформации ориентации. Кинематика с тремя линейными осями и тремя ортогональными круговыми осями. Круговые оси расположены параллельно двум из трех линейных осей. Первая круговая ось приводится в движение двумя декартовыми линейными осями. Она вращает третью линейную ось с инструментом. Вторая круговая ось вращает деталь. Базовая ориентация инструмента может быть запрограммирована посредством дополнительного вращения вокруг себя самой с углом поворота THETA.

При вызове "Базовой 3-, 4- и 5-/6-осевой трансформации" дополнительно может быть передана первичная ориентация инструмента. Ограничения касательно направлений круговых осей более не действуют. Если круговые оси расположены не точно вертикально друг на друге или если имеющиеся круговые оси расположены не точно параллельно линейным осям, то "Базовая 5-/6-осевая трансформация" может обеспечить лучшие результаты ориентации инструмента..

Кинематические трансформации TRANSMIT, TRACYL и TRAANG

Для фрезерных обработок на токарных станках или для подаваемой под наклоном оси при шлифовании, в зависимости от трансформации, в стандартной ситуации действуют следующие расположения осей:

TRANSMIT	Активация полярной трансформации
Торцовая обработка в токарном зажиме	Одна круговая ось Одна ось подачи вертикально к оси вращения Одна продольная ось параллельно оси вращения

TRACYL	Активация трансформации боковой поверхности цилиндра
Обработка пазов произвольной формы на цилиндрических телах	Одна круговая ось Одна ось подачи вертикально к оси вращения Одна продольная ось параллельно оси вращения

TRAANG	Активация трансформации наклонной оси
Обработка с наклонной осью подачи	Одна круговая ось Одна ось подачи с параметризуемым углом Одна продольная ось параллельно оси вращения

Движение “от точки к точке” в декартовой системе координат

Движение станка выполняется в координатах станка и программируется с:

TRAORI	Активация трансформации
PTP Движение "от точки к точке"	Подвод к позиции в декартовой системе координат (MCS)
CP	Движение по траектории декартовых осей в (BKS)
STAT	Положение шарниров зависит от трансформации
TU	На какой угол оси двигаются по кратчайшему пути

Движение от точки к точке при базовой 5/6-осевой трансформации

Движение станка выполняется в координатах станка и ориентация инструмента может быть запрограммирована как с позициями круговых осей, так и с независимыми от кинематики векторами эйлера угла или угла PRY, или векторами направления.

При этом возможна интерполяция круговой оси, векторная интерполяция с большой круговой интерполяцией или интерполяция вектора ориентации на боковой поверхности конуса.

Пример 3- до 5-осевой трансформации для карданной фрезерной головки

Станок имеет минимум 5 осей, из которых

- Три линейные оси для прямолинейных движений, перемещающие рабочую точку в любую позицию в рабочей зоне.
- Две круговые оси качаний, расположенные под сконфигурированным углом (в большинстве случаев 45 градусов), позволяют инструменту принимать ориентации в пространстве, которые при расположении под углом 45 градусов ограничиваются полусферой.

7.1.2 Обзор трансформации ориентации TRAORI

Возможные типы программирования в комбинации с TRAORI

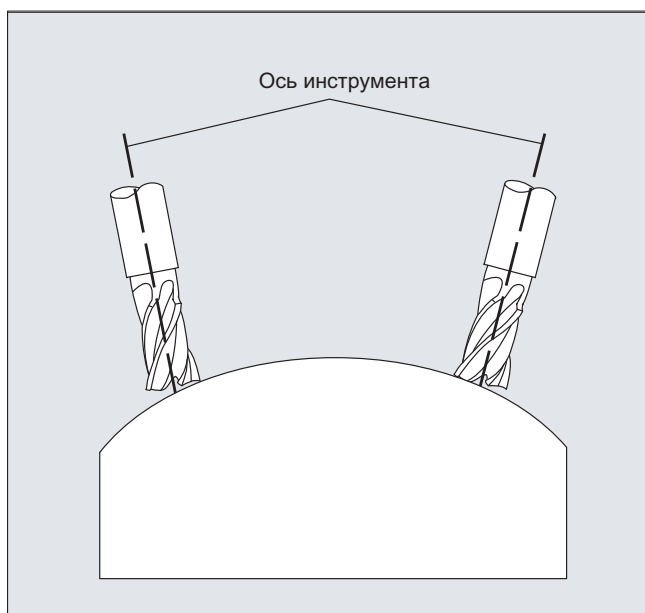
Тип станка	Программирование при активной трансформации TRAORI
Типы станков 1, 2 или 3 Двухосевая качающаяся головка или двухосевой поворотный стол или комбинация одноосевой качающейся головки и поворотного стола соответственно.	Последовательность осей ориентации и направление ориентации инструмента может проектироваться либо относительно станка через машинные данные в зависимости от кинематики станка, либо относительно детали с программируемой ориентацией независимо от кинематики станка Направления вращения осей ориентации в базовой системе программируются с: - ORIMKS базовая система = система координат станка - ORIWKS базовая система = система координат детали Первичной установкой является ORIWKS. Программирование осей ориентации с: - A, B, C позиций осей станка напрямую - A2, B2, C2 программирование угла виртуальных осей с - ORIEULER через эйлеров угол (стандарт) - ORIRPY через угол RPY - ORIVIRT1 через виртуальные оси ориентации 1-ого определения - ORIVIRT2 через виртуальные оси ориентации 2-ого определения с дифференциацией типа интерполяции: линейная интерполяция - ORIAxes осей ориентации или осей станка большая круговая интерполяция (интерполяция вектора ориентации) - ORIVECT осей ориентации Программирование осей ориентации через указание A3, B3, C3 компонентов вектора (направления/нормали плоскости) Программирование результирующей ориентации инструмента A4, B4, C4 вектора нормалей плоскостей в начале кадра A5, B5, C5 вектора нормалей плоскостей в конце кадра LEAD угол предварения для ориентации инструмента TILT боковой угол для ориентации инструмента

Тип станка	Программирование при активной трансформации TRAORI
	Интерполяция вектора ориентации на боковой поверхности конуса Изменения ориентации на расположенной произвольно в пространстве боковой поверхности конуса через интерполяцию: - ORIPLANE в плоскости (большая круговая интерполяция) - ORICONCW на боковой поверхности конуса по часовой стрелке - ORICONCCW на боковой поверхности конуса против часовой стрелки A6, B6, C6 вектора направления (ось вращения конуса) -OICONIO интерполяция на боковой поверхности конуса с: A7, B7, C7 промежуточные векторы (стартовая и конечная ориентация) или - ORICONTO на боковой поверхности конуса тангенциальный переход Изменения ориентации относительно траектории с - ORICURVE задача движения двух контактных точек через PO[XH]=(xe, x2, x3, x4, x5) Полиномы ориентации до 5-ого порядка PO[YH]=(ye, y2, y3, y4, y5) Полиномы ориентации до 5-ого порядка PO[ZH]=(ze, z2, z3, z4, z5) Полиномы ориентации до 5-ого порядка - ORIPATHS Сглаживание характеристики ориентации с A8, B8, C8 Этап переориентации инструмента соответствует: направление и длина хода инструмента при движении отвода
Типы станков 1 и 3 Для других типов станков с дополнительным вращением инструмента вокруг себя самого необходима третья круговая ось Трансформация ориентации, к примеру, базовая 6-осевая трансформация. Вращение вектора ориентации.	Программирование вращений ориентации инструмента с LEAD Угол предварения относительно вектора нормалей плоскостей PO[PHI] Программирование полинома макс. 5-ого порядка TILT Боковой угол Вращение вокруг касательной к траектории (направление Z) PO[PSI] Программирование полинома макс. 5-ого порядка THETA Угол поворота (вращение вокруг направления инструмента в Z) THETA= значение, достигаемое в конце кадра THETA=AC(...) Покадровое переключение на абсолютное указание размера THETA=IC(...) Покадровое переключение на указание составного размера THETA=Θ. Интерполяция запрограммированного угла G90/G91 PO[THT]=(...) Программирование полинома макс. 5-ого порядка Программирование вектора вращения - ORIOTA абсолютное вращение - ORIOTR относительный вектор вращения - ORIOTT тангенциальный вектор вращения
Ориентация относительно траектории для изменений ориентации относительно траектории или вращения вектора вращения тангенциально к траектории	Изменения ориентации относительно траектории с - ORIPATH Ориентация инструмента относительно траектории - ORIPATHS дополнительно при изломе характеристики ориентации Программирование вектора вращения - ORIOTC тангенциальный вектор вращения, вращение к касательной к траектории

7.2 3-, 4- и 5-осевая трансформация (TRAORI)

7.2.1 Общие связи карданной инструментальной головки

Для достижения оптимальных условий резания при обработке изогнутых в пространстве поверхностей требуется возможность изменения угла установки инструмента.

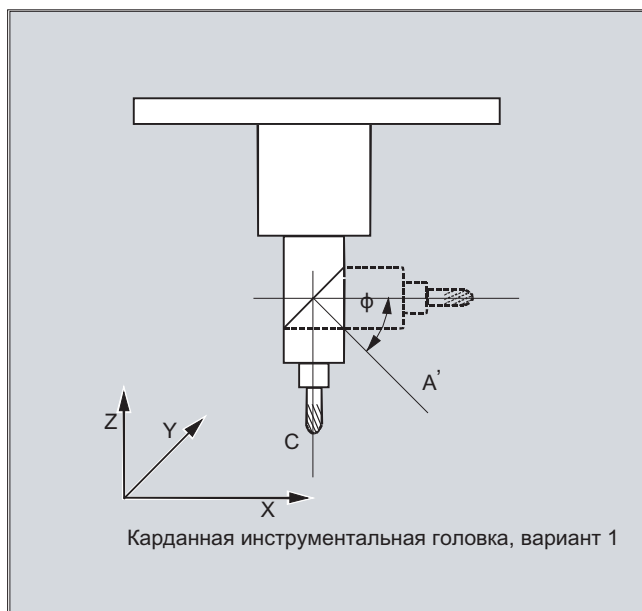


С какой конструкцией станка это достигается, зафиксировано в данных осях.

5-осевая трансформация

Карданная инструментальная головка

Здесь три линейные оси (X, Y, Z) и две оси ориентации (C, A) определяют угол установки и рабочую точку инструмента. Одна из двух осей ориентации является наклонной осью, здесь в примере A' – во многих случаях располагается под углом 45°.



В приведенных здесь примерах показаны расположения на примере с карданной инструментальной головкой кинематики станка CA!

Изготовитель станка

Последовательность круговых осей и направление ориентации инструмента могут устанавливаться в зависимости от кинематики станка через машинные данные.



В этом примере A' лежит под углом ϕ к оси X

В общем и целом действуют следующие возможные связи:

A' лежит под углом φ к оси X

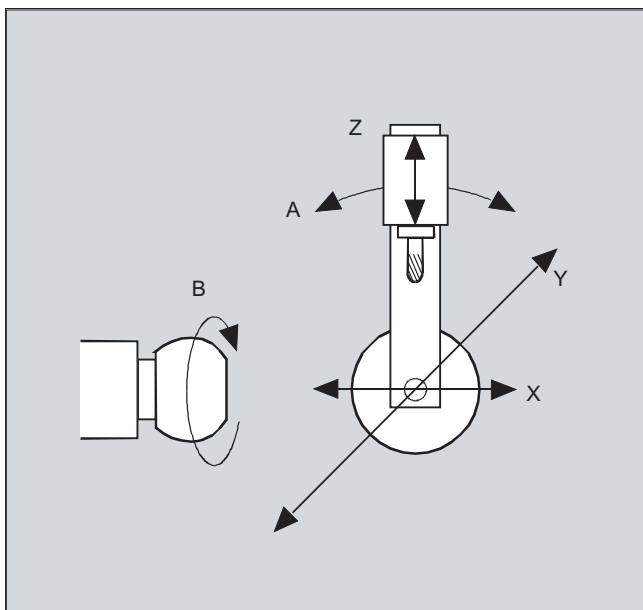
B' лежит под углом φ к оси Y

C' лежит под углом φ к оси Z

Угол φ может быть сконфигурирован в диапазоне 0° до $+89^\circ$ через машинные данные.

С качающейся линейной осью

Здесь речь идет о компоновке с подвижной деталью и подвижным инструментом. Кинематика состоит из трех линейных осей (X, Y, Z) и двух расположенных под прямым углом осей вращения. Первая круговая ось приводится в движение, к примеру, через крестовый суппорт двумя линейными осями, инструмент располагается параллельно третьей линейной оси. Вторая ось вращения поворачивает деталь. Третья линейная ось (ось качания) лежит в плоскости крестового суппорта.



Последовательность круговых осей и направление ориентации инструмента могут устанавливаться в зависимости от кинематики станка через машинные данные.

Действуют следующие возможные связи:

Оси:	Последовательности осей:
1. Круговая ось	A A B B C C
2. круговая ось	B C A C A B
Повернутая линейная ось	Z Y Z X Y X

Прочие пояснения по конфигурируемым последовательностям осей для направления ориентации инструмента см.

Литература: /FB3/ Описание функций "Специальные функции"; 3- до 5-осевые трансформации (F2), глава "Карданная фрезерная головка", "Параметрирование".

7.2.2 3-, 4- и 5-осевая трансформация (TRAORI)

Пользователь может сконфигурировать две или три линейные оси и одну круговую ось. Трансформации исходят из того, что круговая ось расположена ортогонально на плоскости ориентации.

Ориентация инструмента возможна только в плоскости, расположенной вертикально к круговой оси. Трансформация поддерживает типы станков с подвижным инструментом и подвижной деталью.

Проектирование и программирование 3- и 4-осевых трансформаций аналогично таковым для 5-осевой трансформации.

Литература:

Описание функций "Специальные функции"; Многоосевые трансформации (F2)

Синтаксис

TRAORI (<n>)

TRAORI (<n>, <X>, <Y>, <Z>, <A>,)

TRAFOOF

Значение

TRAORI:	Активирует первую согласованную трансформацию ориентации	
TRAORI (<n>):	Активирует согласованную с n трансформацию ориентации	
<n>:	номер трансформации	
	Значение:	1 или 2
	Пример: TRAORI(1) активирует трансформацию ориентации 1	
<X>, <Y>, <Z>:	Компонент вектора ориентации, на который указывает инструмент.	
<A>, :	Программируемое смещение для круговых осей	
TRAFOOF:	Отключить трансформацию	

Ориентация инструмента

В зависимости от выбранного направления ориентации инструмента, в программе ЧПУ активная рабочая плоскость (G17, G18, G19) должна быть установлена таким образом, чтобы коррекция длин инструмента действовала в направлении ориентации инструмента.

Примечание

После включения трансформации данные позиций (X, Y, Z) всегда относятся к острию инструмента. Изменение позиций участвующих в трансформации круговых осей приводит к движениям компенсации прочих осей станка, из-за чего позиция острия инструмента остается неизменной.

Трансформация ориентации всегда направлена от острия к зажиму инструмента.

Смещение для осей ориентации

При активации трансформации ориентации можно напрямую запрограммировать дополнительное смещение для осей ориентации.

Можно пропустить параметры, если при программировании соблюдается правильная последовательность.

Пример:

TRAORI (, , , , A, B) : если необходимо ввести только одно единственное смещение.

В качестве альтернативы прямому программированию, дополнительное смещение для осей ориентации может быть автоматически взято и из актуального активного смещения нулевой точки. Передача конфигурируется через машинные данные.

Примеры

TRAORI (1, 0, 0, 1)	; Первичная ориентация инструмента указывает в направлении Z
TRAORI (1, 0, 1, 0)	; Первичная ориентация инструмента указывает в направлении Z
TRAORI (1, 0, 1, 1)	; Первичная ориентация инструмента указывает в направлении Y/Z (соответствует позиции -45°)

7.2.3 Варианты программирования ориентации и первичная установка (ORIRESET)

Программирование ориентации инструмента при TRAORI

В комбинации с программируемой трансформацией ориентации TRAORI дополнительно к линейным осям X, Y, Z через идентификаторы круговых осей A..., B..., C... могут быть запрограммированы позиции осей или виртуальные оси с углами или компоненты вектора. Для осей ориентации и осей станка возможны различные типы интерполяции. Независимо от того, какие полиномы ориентации PO[угол] и полиномы осей PO[ось] активны на соответствующий момент, может быть запрограммировано несколько различных типов полиномов, например, G1, G2, G3, CIP или POLY.

Изменение ориентации инструмента может быть запрограммировано и через векторы ориентации. При этом конечная ориентация каждого кадра может быть выполнена либо через прямое программирование вектора, либо через программирование позиций круговых осей.

Варианты программирования ориентации при 3- до 5-осевой трансформации

Следующие варианты программирования ориентации являются взаимоисключающими.

A, B, C	Прямое указание позиции круговой оси
A2, B2, C2	Программирование угла виртуальных осей через угол Эйлера или угол RPY.
A3, B3, C3	Указание компонентов вектора
LEAD, TILT	Указание угла предварения и бокового угла относительно траектории и поверхности

A4, B4, C4 A5, B5, C5	вектор нормали плоскости в начале кадра и в конце кадра
A6, B6, C6 A7, B7, C7	Интерполяция вектора ориентации на боковой поверхности конуса
A8, B8, C8	Переориентация инструмента, направления и длины хода инструмента при движении отвода

Подвод к первичной установке ориентации инструмента

Посредством `ORIRESET(...)` оси ориентации соответствующей кинематики станка линейно и синхронно перемещаются из своей текущей позиции на запрограммированную позицию первичной установки. Если позиция первичной установки для оси запрограммирована, то используется позиция из соответствующих машинных данных `$MC_TRAFO5_ROT_AX_OFFSET_1/2`.

Активные фреймы круговых осей не учитываются.

Примеры кинематики станка CA (имена осей канала C, A)

Команды	Описание
<code>ORIRESET(90, 45)</code>	Ось C: 90° Ось A: 45°
<code>ORIRESET(, 30)</code>	Ось C: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[0]</code> Ось A: 30°
<code>ORIRESET()</code>	Ось C: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[0]</code> Ось A: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[1]</code>

Примеры кинематики станка SAC (имена осей канала C, A, B)

Команды	Описание
<code>ORIRESET(90, 45, 90)</code>	Ось C: 90° Ось A: 45° Ось B: 90°
<code>ORIRESET()</code>	Ось C: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[0]</code> Ось A: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[1]</code> Ось B: <code>\$MC_TRAFO5_ROT_AX_OFFSET_1/2[2]</code>

Примечание

Подвод к исходному положению ориентации инструмента с помощью функции `ORIRESET(...)` разрешено выполнять, только если трансформация ориентации `TRAORI(...)` активна.

Программирование вращений LEAD, TILT и THETA

Угол предварения LEAD и боковой угол TILT

Вращения ориентации инструмента программируются при 3- до 5-осевой трансформации с углом предварения LEAD и боковым углом TILT.

Угол поворота THETA

При трансформации с **третьей круговой осью** можно запрограммировать вращение инструмента вокруг себя самого с углом поворота THETA для ориентирования как с компонентами вектора, так и с программированием углов LEAD, TILT.

7.2.4 Программирование ориентации инструмента (A..., B..., C..., LEAD, TILT)

Для программирования ориентации инструмента существуют следующие возможности:

1. Прямое программирование движения круговых осей. Изменение ориентации всегда выполняется в базовой кинематической или системе координат станка. Оси ориентации перемещаются как синхронные оси.
2. Программирование в углах Эйлера или RPY согласно определению угла через A2, B2, C2.
3. Программирование вектора направления через A3, B3, C3. Вектор направления указывает от острия в направлении зажима инструмента.
4. Программирование вектора нормалей плоскостей в начале кадра с A4, B4, C4 и в конце кадра с A5, B5, C5 (торцовое фрезерование).
5. Программирование через угол предварения LEAD и боковой угол TILT
6. Программирование оси вращения конуса как нормированного вектора через A6, B6, C6 или промежуточной ориентации на боковой поверхности конуса через A7, B7, C7, см. главу "Программирование ориентации вдоль боковой поверхности конуса (ORIPANE, ORICONxx)".
7. Программирование переориентации, направления и длины хода инструмента при движении отвода через A8, B8, C8, см. главу "Сглаживание характеристики ориентации (ORIPATHS A8=, B8=, C8=)"

Примечание

Во всех случаях программирование ориентации разрешено только тогда, когда включена трансформация ориентации.

Преимущество: Эти программы работают на любой кинематике станка.

Определение ориентации инструмента через команду G

Примечание

Изготовитель станка

Через машинные данные можно переключаться между углами Эйлера или RPY. При соответствующих настройках машинных данных возможно переключение, как зависимое, так и не зависимое от активной команды G группы 50. На выбор имеются следующие возможности настройки:

1. Если оба машинных параметра для определения осей ориентации и определения угла ориентации установлены G-командой на ноль: Запрограммированные с A2, B2, C2 углы, **независимо от машинных данных** определения углов программирования ориентации, интерпретируются либо как углы Эйлера, либо как углы RPY.
2. Если машинный параметр определения осей ориентации через G-команду установлен на единицу, то переключение осуществляется **в зависимости** от активной G-команды группы 50:
Запрограммированные с A2, B2, C2 углы интерпретируются согласно одной из активных G-команд ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS и ORIPY2. Запрограммированные с осями ориентации значения согласно активной G-команде группы 50 интерпретируются и как угол ориентации.
3. Если машинный параметр для определения угла ориентации через G-команду установлен на единицу, а машинный параметр для определения осей ориентации через G-команду установлен на ноль, то переключение осуществляется **независимо** с активной G-команды группы 50:
Запрограммированные с A2, B2, C2 углы интерпретируются согласно одной из активных G-команд ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS и ORIPY2. Независимо от активной G-команды группы 50 запрограммированные с осями ориентации значения всегда интерпретируются как позиции круговых осей.

Синтаксис

Позиции круговых осей

G1 X<значение> Y<значение> Z<значение> A<значение> B<значение> C<значение>

Угол Эйлера

G1 X<значение> Y<значение> Z<значение> A2=<значение> B2=<значение>
C2=<значение>

Вектор направления

G1 X<значение> Y<значение> Z<значение> A3=<значение> B3=<значение>
C3=<значение>

Вектор нормали к плоскости в начале кадра

G1 X<значение> Y<значение> Z<значение> A4=<значение> B4=<значение>
C4=<значение>

Вектор нормали к плоскости в конце кадра

G1 X<значение> Y<значение> Z<значение> A5=<значение> B5=<значение>
C5=<значение>

Угол предварения

LEAD=<значение>

Боковой угол

TILT=<значение>

Объяснение

G1:	Линейная интерполяция
X, Y, Z:	Позиции линейных осей
A, B, C:	Позиции круговых осей
A2=, B2=, C2=:	Программирование угла (угол Эйлера или угол RPY)
A3=, B3=, C3=:	Вектор направления в координатах X, Y, Z в WCS
A4=, B4=, C4=:	Вектор нормали плоскости в начале кадра в координатах X, Y, Z в WCS
A5=, B5=, C5=:	Вектор нормали плоскости в конце кадра в координатах X, Y, Z в WCS
LEAD= :	Угол предварения ¹⁾
TILT= :	Боковой угол ¹⁾
1) Интерпретация указаний углов зависит от предустановки в MD21094 \$MC_ORIPATH_MODE	

Дополнительная информация

Как правило, 5-осевые программы создаются системами CAD/CAM, а не вводятся на СЧПУ. Поэтому объяснения ниже прежде всего предназначены для программистов постпроцессоров.

Для программирования ориентации доступны следующие команды:

Команда	Объяснение
ORIEULER:	Угол Эйлера с последовательностью вращений ZX'Z"
ORIRPY:	Угол RPY с последовательностью вращений XY'Z"
ORIRPY2:	Угол RPY с последовательностью вращений ZY'X"
ORIVIRT1:	Виртуальные оси ориентации со свободно определяемой последовательностью вращений через: MD21120 \$MC_ORIAX_TURN_TAB_1
ORIVIRT2:	Виртуальные оси ориентации со свободно определяемой последовательностью вращений через: MD21130 \$MC_ORIAX_TURN_TAB_2
ORIAPOS:	Виртуальные оси ориентации с позициями круговых осей

Примечание

Через машинные данные изготовителем станка могут быть определены различные варианты. Следовать указаниям изготовителя станка.

Программирование в углах Эйлера ORIEULER, последовательность вращений Z X' Z''

Значения, запрограммированные при программировании ориентации ORIEULER с A2, B2, C2, интерпретируются как угол Эйлера (в градусах).

Новый вектор ориентации получается посредством следующих трех вращений изначального вектора ориентации

1. с круговой осью A2 вокруг оси координат Z
2. с круговой осью B2 вокруг новой оси координат X'
3. с круговой осью C2 вокруг новой оси координат Z''

В этом случае значение C2 (поворот вокруг новой оси Z) не имеет значения, и его программирование не требуется.

Программирование в углах RPY ORIRPY ORIRPY, последовательность вращений X Y' Z''

Значения, запрограммированные при программировании ориентации ORIRPY посредством A2, B2, C2, интерпретируются как угол RPY (в градусах) с последовательностью вращений X Y' Z'.

Примечание

В отличие от программирования посредством ORIEULER, при ORIRPY все три значения влияют на вектор ориентации.

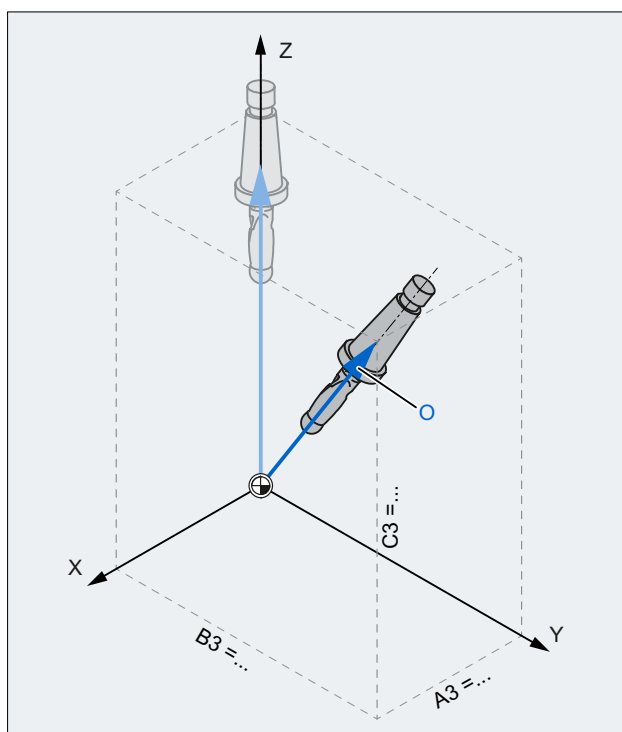
Новый вектор ориентации получается посредством следующих трех вращений изначального вектора ориентации

1. с круговой осью A2 вокруг оси координат X
2. с круговой осью B2 вокруг новой оси координат Y'
3. с круговой осью C2 вокруг новой оси координат Z''

Программирование вектора направления

Компоненты вектора направления программируются с A3, B3, C3. Вектор указывает в направлении зажима инструмента; длина вектора при этом не имеет значения.

Не запрограммированные компоненты вектора устанавливаются на ноль.



X, Y, Z Оси координат WCS

A3, B3, C3 Компоненты вектора направления

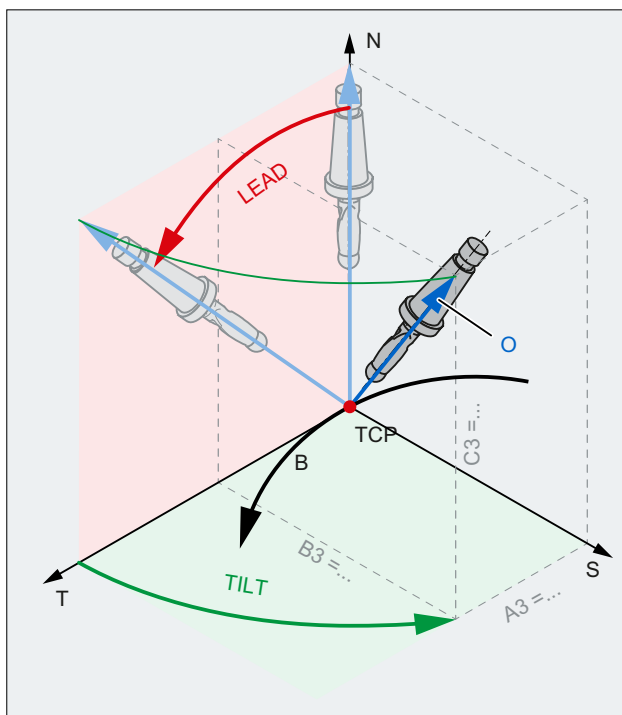
O вектор ориентации

Рисунок 7-1 Программирование вектора направления

Программирование ориентации инструмента посредством LEAD и TILT

Результирующая ориентация инструмента получается из:

- касательной к траектории
- вектора нормали к плоскости
В начале кадра A4, B4, C4 и в конце кадра A5, B5, C5
- Угол предварения LEAD
Угол в плоскости, образованной касательной к траектории и вектором нормали к плоскости
- Боковой угол TILT в конце кадра
Угол в плоскости, вертикально к касательной к траектории относительно вектора нормали к плоскости



- T Касательная к траектории
- S Перпендикуляр к касательной к траектории.
- N Нормаль к плоскости
- B Траектория
- TCP Tool Center Point (центр инструмента)
- O вектор ориентации

Рисунок 7-2 Программирование LEAD TILT

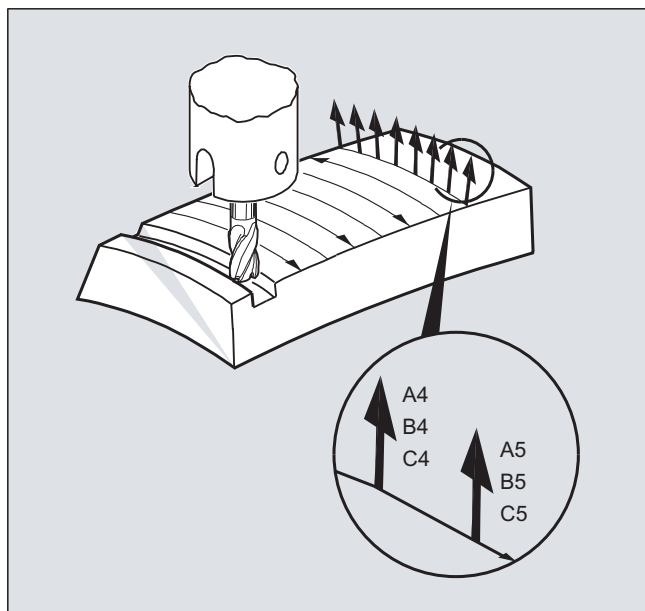
Примечание

Поведение на внутренних углах при 3D-коррекции инструмента (WZK)

Даже если кадр на внутреннем углу укорачивается, запрограммированная ориентация инструмента на конце кадра все равно устанавливается.

7.2.5 Торцовое фрезерование (A4, B4, C4, A5, B5, C5)

Торцовое фрезерование служит для обработки поверхностей с любым изгибом.



Для этого типа фрезерования 3D необходимо построочное описание траекторий 3D на поверхности детали.

Вычисления с учетом формы и размеров инструмента обычно выполняются в САМ. После полностью вычисленные кадры ЧПУ через постпроцессоры загружаются в СЧПУ.

Программирование искривления траектории

Описание поверхностей

Описание искривления траектории осуществляется через векторы нормалей плоскостей со следующими компонентами:

A4, B4, C4 Стартовый вектор в начале кадра

A5, B5, C5 Конечный вектор на конце кадра

Если в кадре стоит только стартовый вектор, то вектор нормали плоскости остается постоянным во всем кадре. Если в кадре стоит только конечный вектор, то от конечного значения предшествующего кадра через большую круговую интерполяцию выполняется интерполяция к запрограммированному конечному значению.

Если запрограммированы стартовый и конечный вектор, то между двумя направлениями также выполняется большая круговая интерполяция. Благодаря этому достигаются непрерывные ровные ходы траектории.

В первичной установке векторы нормалей плоскостей, независимо от активной плоскости G17 до G19, указывают в направлении Z.

Длина вектора не имеет значения.

Не запрограммированные компоненты вектора устанавливаются на ноль.

При активной ORIWKS (см. "Нулевая точка осей ориентации (ORIWKS, ORIMKS) (Страница 356)") векторы нормалей плоскостей относятся к активному фрейму и при вращении фрейма также поворачиваются.

Изготовитель станка

Вектор нормали плоскости должен располагаться в пределах настраиваемого через машинные данные граничного значения вертикально к касательной к траектории, иначе выводится ошибка.

7.2.6 Нулевая точка осей ориентации (ORIWKS, ORIMKS)

При программировании ориентации в системе координат детали через

- угол Эйлера или угол RPY или
- вектор ориентации

можно установить ход вращательного движения через ORIMKS/ORIWKS.

Примечание

Изготовитель станка

Тип интерполяции для ориентации устанавливается с помощью машинных данных:

MD21104 \$MC_ORI_IPO_WITH_G_CODE

=FALSE: Базой являются G-команды ORIWKS и ORIMKS

=TRUE: Базой являются G-команды 51-й группы (ORIAXES, ORIVECT, ORIPLANE, ...)

Синтаксис

ORIMKS=...

ORIWKS=...

Объяснение

ORIMKS:	Вращение в системе координат станка
ORIWKS:	Вращение в системе координат детали

Примечание

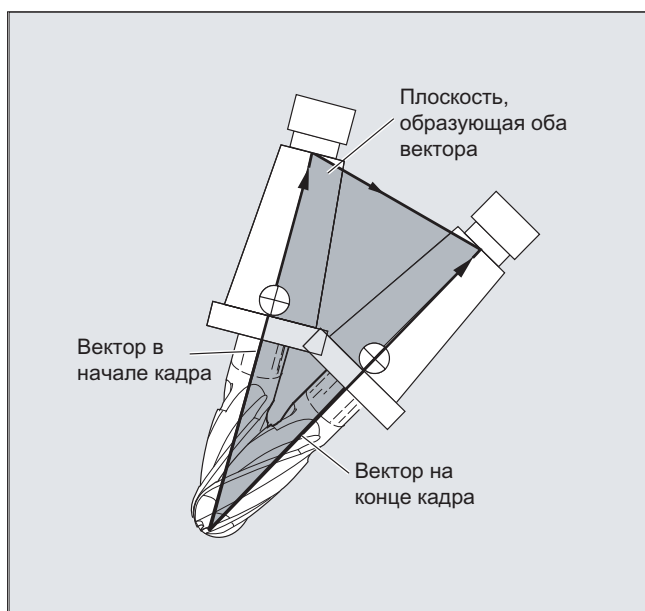
ORIWKS является первичной настройкой. Если в случае 5-осевой программы изначально не ясно, на каком станке она должна работать, то всегда выбирается ORIWKS. Какие движения в действительности выполняет станок, зависит от его кинематики.

С помощью ORIMKS можно запрограммировать фактические движения станка, например, во избежание столкновений с приспособлениями или и т. п.

Дополнительная информация

При **ORIMKS** выполняемое движение инструмента **зависит** от кинематики станка. При изменении ориентации с зафиксированным в пространстве острием инструмента выполняется линейная интерполяция между позициями круговых осей.

При **ORIWKs** движение инструмента **не зависит** от кинематики станка. При изменении ориентации с зафиксированным в пространстве острием инструмента, инструмент движется в образованной начальным и конечным вектором плоскости.



Сингулярные позиции

Примечание

ORIWKs

Для движений ориентации в области сингулярной позиции 5-осевого станка требуются большие движения осей станка (к примеру, в случае поворотной токарной головки с C в качестве оси вращения и A в качестве оси качания все позиции с $A=0$ являются сингулярными).

Изготовитель станка

Во избежание перегрузки осей станка, управление скоростью значительно снижает скорость движения по траектории вблизи от сингулярных позиций.

С помощью машинных данных

```
$MC_TRAFO5_NON_POLE_LIMIT
```

```
$MC_TRAFO5_POLE_LIMIT
```

трансформацию можно спараметризовать так, что движения ориентации вблизи полюса прокладываются через полюс, и возможна быстрая обработка.

Сингулярные позиции обрабатываются только с MD `$MC_TRAFO5_POLE_LIMIT`.

Литература:

/FB3/ Описание функций "Специальные функции"; 3- до 5-осевая трансформация (F2), глава "Сингулярные позиции и их обработка".

7.2.7 Программирование осей ориентации (ORIAxes, ORIVect, ORIEuler, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

Функция "Оси ориентации" описывает ориентацию инструмента в пространстве и достигается через программирование смещений для круговых осей. Следующая третья степень свободы может быть достигнута через дополнительное вращение инструмента вокруг себя самого. Эта ориентация инструмента выполняется произвольно в пространстве через третью круговую ось и для нее необходима 6-осевая трансформация. Самовращение инструмента вокруг себя самого устанавливается в зависимости от типа интерполяции векторов вращения с углом поворота THETA (см. "Вращения ориентации инструмента (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Страница 368)".

Оси ориентации программируются через идентификаторы осей A2, B2, C2.

Синтаксис

N... ORIAxes/ORIVect ; линейная или большая круговая интерполяция
N... G1 X Y Z A B C

N... ORIPlane ; интерполяция ориентации плоскости

N... ORIEuler/ORIRPY/ORIRPY2 ; угол ориентации Эйлера/RPY
N... G1 X Y Z A2 = B2 = C2 = ; программирование угла виртуальных осей

N... ORIVIRT1/ORIVIRT2 ; виртуальные оси ориентации опр. 1/2
N... G1 X Y Z A3 = B3 = C3 = ; программирование вектора направления

Примечание

Для изменений ориентации находящейся в пространстве боковой поверхности конуса могут быть запрограммированы и другие смещения круговой оси осей ориентации (см. "Программирование ориентации вдоль боковой поверхности конуса (ORIPlane, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Страница 360)".

Значение

ORIAxes:	Линейная интерполяция осей станка или осей ориентации
ORIVect:	Большая круговая интерполяция (идентично ORIPlane)

ORIMKS: ORIWKS:	Вращение в системе координат станка Вращение в системе координат детали Описание см. "Нулевая точка осей ориентации (ORIWKS, ORIMKS) (Страница 356)".
A= B= C=:	Программирование позиции оси станка
ORIEULER:	Программирование ориентации через угол Эйлера
ORIRPY:	Программирование ориентации через угол RPY Последовательность вращения XYZ, при этом действует: • A2 это угол поворота вокруг X • B2 это угол поворота вокруг Y • C2 это угол поворота вокруг Z
ORIRPY2:	Программирование ориентации через угол RPY Последовательность вращения ZYX, при этом действует: • A2 это угол поворота вокруг Z • B2 это угол поворота вокруг Y • C2 это угол поворота вокруг X
A2= B2= C2=:	Программирование угла виртуальных осей
ORIVIRT1/ORIVIRT2:	Программирование ориентации через виртуальные оси ориентации Определение 1: установка по MD21120 \$MC_ORIAX_TURN_TAB_1 Определение 2: установка по MD21130 \$MC_ORIAX_TURN_TAB_2
A3= B3= C3=:	Программирование вектора направления оси направления

Дополнительная информация

Изготовитель станка

Посредством MD21102 \$MC_ORI_DEF_WITH_G_CODE устанавливается, как определяются запрограммированные углы A2, B2, C2:

определение выполняется по MD21100 \$MC_ORIENTATION_IS_EULER (стандарт) или определение выполняется по G-группе 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Посредством MD21104 \$MC_ORI_IPO_WITH_G_CODE устанавливается, какой тип интерполяции действует: ORIWKS/ORIMKS или ORIAXES/ORIVECT.

Режим работы JOG

В этом режиме работы всегда выполняется линейная интерполяция углов ориентации. При непрерывном и инкрементальном перемещении клавишами перемещения может перемещаться только одна ось ориентации. Маховичками оси ориентации можно перемещать одновременно.

Для перемещения осей ориентации вручную действует спец. для канала переключатель процентовки подачи или переключатель процентовки ускоренного хода при наложении ускоренного хода.

Следующие машинные данные обеспечивают отдельную задачу скорости:

MD21160 \$MC_JOG_VELO_RAPID_GEO

MD21165 \$MC_JOG_VELO_GEO

MD21150 \$MC_JOG_VELO_RAPID_ORI

MD21155 \$MC_JOG_VELO_ORI

Примечание

SINUMERIK 840D sl с "Пакетом трансформаций для манипуляторов"

С помощью функции "Декартово перемещение вручную" в режиме JOG возможно раздельное преобразование поступательного движения гео-осей в базовых системах MCS, WCS и TCS.

Литература:

Описание функций "Дополнительные функции"; Кинематическая трансформация (M1)

7.2.8

Программирование ориентации вдоль боковой поверхности конуса (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

С помощью расширенной ориентации можно выполнять изменения ориентации вдоль находящейся в пространстве боковой поверхности конуса. Интерполяция вектора ориентации на боковой поверхности конуса осуществляется с помощью модальных команд ORICONxx. Для интерполяции в одной плоскости можно запрограммировать конечную ориентацию с ORIPLANE. В общем и целом стартовая ориентация определяется предшествующими кадрами.



Программирование

Конечная ориентация определяется либо через указание программирования угла в углах Эйлера или RPY с A2, B2, C2, либо через программирование позиций круговых осей с A, B, C. Для осей ориентации вдоль боковой поверхности конуса требуются дополнительные данные программирования:

- Ось вращения конуса как вектор с A6, B6, C6
- Апертурный угол PSI с идентификатором NUT
- Промежуточная ориентация на боковой поверхности конуса с A7, B7, C7

Примечание

Программирование вектора направления A6, B6, C6 для оси вращения конуса

Программирование конечной ориентации не является обязательным. Если конечная ориентация не указана, то выполняется интерполяция всей боковой поверхности конуса с 360 градусами.

Программирование апертурного угла конуса с NUT=угол

Обязательно указать конечную ориентацию.

Интерполяция всей боковой поверхности конуса с 360 градусами таким образом невозможна.

Программирование промежуточной ориентации A7, B7, C7 на боковой поверхности конуса

Обязательно указать конечную ориентацию. Изменение ориентации и направление вращения однозначно определяются через три вектора стартовой, конечной и промежуточной ориентации. При этом все три вектора должны отличаться друг от друга. Если запрограммированная промежуточная ориентация параллельна стартовой или конечной ориентации, то выполняется линейная большая круговая интерполяция в плоскости, образуемой стартовым и конечным вектором.

Расширенная интерполяция ориентации на боковой поверхности конуса

```
N... ORICONCW или ORICONCCW
N... A6= B6= C6= A3= B3= C3=
или
N... ORICONTO
N... G1 X Y Z A6= B6= C6=
или
N... ORICONIO
N... G1 X Y Z A7= B7= C7=
N... PO[PHI]=(a2, a3, a4, a5)
N... PO[PSI]=(b2, b3, b4, b5)
```

Интерполяция на боковой поверхности конуса с

вектором направления по/против часовой стрелки конуса и конечной ориентацией или

тангенциальным переходом и указанием конечной ориентации

или

указанием конечной ориентации с промежуточной ориентацией на боковой поверхности конуса с

полиномом для угла поворота с полиномом для апертурного угла

Параметры

ORIPANE:	Интерполяция в плоскости (большая круговая интерполяция)
ORICONCW:	Интерполяция на боковой поверхности конуса по часовой стрелке
ORICONCCW:	Интерполяция на боковой поверхности конуса против часовой стрелки
ORICONTO:	Интерполяция на боковой поверхности конуса, тангенциальный переход
A6= B6= C6=:	Программирование оси вращения конуса (нормированный вектор)
NUT=угол:	Апертурный угол конуса в градусах
NUT=+179:	Угол перемещения меньше или равен 180 градусам
NUT=-181:	Угол перемещения больше или равен 180 градусам
ORICONIO:	Интерполяция на боковой поверхности конуса
A7= B7= C7=:	Промежуточная ориентация (программирование как нормированный вектор)
PHI:	Угол поворота ориентации вокруг оси направления конуса
PSI:	Апертурный угол конуса
Возможные полиномы PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	Кроме соответствующих углов могут быть запрограммированы и полиномы макс. 5-го порядка

Пример: Различные изменения ориентации

Программный код	Комментарий
...	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	; Трансформация ориентации вкл.
N30 ORIVECT	; Интерполяция ориентации инструмента как вектора.
...	; Ориентация инструмента в плоскости.
N40 ORIPANE	; Выбрать большую круговую интерполяцию.
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; Ориентация в плоскости Y/Z повернута на 45 градусов, на конце кадра достигается ориентация $(0, 1/\sqrt{2}, 1/\sqrt{2})$.
...	
N70 ORICONCW	; Программирование ориентации на боковой поверхности конуса:
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	; Выполняется интерполяция вектора ориентации на боковой поверхности конуса с направлением $(0, 0, 1)$ до ориентации $(1/\sqrt{2}, 0, 1/\sqrt{2})$ по часовой стрелке, угол поворота при этом составляет 270 градусов.
N90 A6=0 B6=0 C6=1	; Ориентация выполняет полный оборот на той же боковой поверхности конуса.

Дополнительная информация

Если необходимо описать изменения ориентации на расположенной произвольно в пространстве боковой поверхности конуса, то должен быть известен вектор, вокруг которого должна вращаться ориентация инструмента. Кроме этого, должны быть заданы начальная и конечная ориентации. Стартовая ориентация следует из предшествующего кадра, а конечная ориентация должна быть либо запрограммирована, либо определена через иные условия.

Программирование в плоскости ORIPLANE соответствует ORIVECT

Программирование большой круговой интерполяции в комбинации с угловыми полиномами соответствует линейной и полиномиальной интерполяции контуров. Интерполяция ориентации инструмента выполняется в плоскости, образованной стартовой и конечной ориентацией. При программировании дополнительных полиномов вектор ориентации может быть опрокинут из плоскости.

Программирование окружностей в плоскости G2/G3, CIP и CT

Расширенная ориентация соответствует интерполяции окружностей в плоскости. По соответствующим возможностям программирования окружностей с указанием центра или радиуса как G2/G3, окружность через промежуточную точку CIP и тангенциальные окружности CT см.

Литература: Руководство по программированию "Основы", "Программирование команд перемещения"..

Программирование ориентации

Интерполяция вектора ориентации на боковой поверхности конуса ORICONxx

Для интерполяции ориентаций на боковой поверхности конуса можно выбрать четыре различных типа интерполяции из группы команд G 51:

1. Интерполяция на боковой поверхности по часовой стрелке **ORICONCW** с указанием конечной ориентации и направления конуса или апертурного угла. Вектор направления программируется с идентификаторами **A6, B6, C6**, а апертурный угол конуса – с идентификатором **NUT**= диапазон значений в интервале 0 до 180 градусов.
2. Интерполяция на боковой поверхности против часовой стрелки **ORICONCWW** с указанием конечной ориентации и направления конуса или апертурного угла. Вектор направления программируется с идентификаторами **A6, B6, C6**, а апертурный угол конуса – с идентификатором **NUT**= диапазон значений в интервале 0 до 180 градусов.
3. Интерполяция на боковой поверхности конуса **ORICONIO** с указанием конечной ориентации и промежуточной ориентации, которая программируется с идентификаторами **A7, B7, C7**.
4. Интерполяция на боковой поверхности конуса **ORICONTO** с тангенциальным переходом и указанием конечной ориентации. Вектор направления программируется с идентификаторами **A6, B6, C6**.

7.2.9 **Задача ориентации двух контактных точек (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)**

Программирование изменения ориентации через вторую пространственную кривую ORICURVE

Следующей возможностью программирования изменений ориентации является программирование, кроме острия инструмента вдоль пространственной кривой, и движения второй контактной точки инструмента с ORICURVE. Таким образом возможно однозначное определение изменений ориентации инструмента, как и при программировании самого вектора инструмента.

Изготовитель станка

Следовать указаниям изготовителя станка касательно настраиваемых через машинные данные идентификаторов осей для программирования 2-й траектории ориентации инструмента.

Программирование

При этом типе интерполяции для обоих пространственных кривых могут быть запрограммированы точки с G1 или полиномы с POLY. Окружности и эвольвенты запрещены. Дополнительно можно активировать сплайн-интерполяцию с BSPLINE и функцию "Соединение коротких сплайн-кадров".

Литература:

Описание функций "Основные функции"; Режим управления траекторией, точный останов, LookAhead (B1), глава: Соединение коротких сплайн-кадров

Другие типы сплайнов ASPLINE и CSPLINE, а также активация компрессора с COMPON, COMPCURV или COMPCAD запрещены.

Движение двух контактных точек инструмента при программировании полиномов ориентации может задаваться для координат до макс. 5-го порядка.

Расширенная интерполяция ориентации с дополнительной пространственной кривой и полиномами для координат

N... ORICURVE	Указание движения второй контактной точки инструмента и дополнительных полиномов соответствующих координат
N... PO[XH]=(xe, x2, x3, x4, x5)	
N... PO[YH]=(ye, y2, y3, y4, y5)	
N... PO[ZH]=(ze, z2, z3, z4, z5)	

Параметры

ORICURVE	Интерполяция ориентации с задачей движения двух контактных точек инструмента.
XH YH ZH	Идентификатор координат второй контактной точки инструмента дополнительного контура как пространственная кривая

Возможные полиномы	Кроме соответствующих конечных точек, пространственные кривые дополнительно могут программироваться с полиномами.
PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)	
xe, ye, ze	Конечные точки пространственной кривой
xi, yi, zi	Коэффициенты полиномов макс. 5-го порядка

Примечание**Идентификаторы XH YH ZH для программирования 2-й траектории ориентации**

Идентификаторы должны быть выбраны таким образом, чтобы не возникло конфликта с другими идентификаторами линейных осей

осями X Y Z

и круговыми осями, например,

A2 B2 C2 угол Эйлера или угол RPY

A3 B3 C3 векторы направления

A4 B4 C4 или A5 B5 C5 векторы нормалей плоскостей

A6 B6 C6 векторы вращения или A7 B7 C7 координаты промежуточной точки или иными параметрами интерполяции.

PO [THT]	Угол поворота G-команд группы 54, запрограммированных через поворот вектора вращения с THETA
PHI	Угол предварения LEAD
PSI	Боковой угол TILT
THETA	Вращение вокруг направления инструмента в Z
PO [XH]	Координата X исходной точки на инструменте
PO [YH]	Координата Y исходной точки на инструменте
PO [ZH]	Координата Z исходной точки на инструменте

Дополнительная информация

Полиномы ориентации не могут быть запрограммированы

- Если активны сплайн-интерполяции ASPLINE, BSPLINE, CSPLINE.
Полиномы типа 1 для угла ориентации возможны для любого типа интерполяции кроме сплайна, т. е. для линейной интерполяции с ускоренным ходом G00 или с подачей G01 при полиномиальной интерполяции с POLY и при круговой или эвольвентной интерполяции с G02, G03, CIP, CT, INVCW и INCCCW.
Полиномы типа 2 для координат ориентации, напротив, возможны только тогда, когда активна линейная интерполяция с ускоренным ходом G00 или с подачей G01 или полиномиальная интерполяция с POLY.
- Если выполняется интерполяция ориентации посредством осевой интерполяции ORIAXES. В этом случае напрямую могут быть запрограммированы полиномы с PO[A] и PO[B] для осей ориентации A и B.

Полиномы ориентации типа 1 с ORIVECT, ORIPLANE и ORICONxx

При большой круговой интерполяции и интерполяции боковой поверхности конуса с ORIVECT, ORIPLANE и ORICONxx возможны только полиномы ориентации типа 1.

Полиномы ориентации типа 2 с ORICURVE

Если активна интерполяция с дополнительной пространственной кривой ORICURVE, то выполняется интерполяция декартовых компонентов вектора ориентации и возможны полиномы ориентации типа 2.

7.4 Вращения ориентации инструмента (ORIROTA, ORIROTR, ORIROTТ, ORIROTC, THETA)

Если для типов станков с подвижным инструментом необходима возможность изменения ориентации инструмента, то каждый кадр программируется с конечной ориентацией. В зависимости от кинематики станка, можно запрограммировать либо направление ориентации, либо направление вращения вектора ориентации THETA. Для этих векторов вращения могут быть запрограммированы различные типы интерполяции:

- ORIROTA: угол поворота к заданному абсолютно направлению вращения.
- ORIROTR: угол поворота относительно плоскости между стартовой и конечной ориентаций.
- ORIROTТ: угол поворота относительно изменения вектора ориентации.
- ORIROTC: тангенциальный угол поворота к касательной к траектории.

Синтаксис

Только при активном типе интерполяции ORIROTA можно запрограммировать угол поворота или вектор вращения на четыре возможных способа следующим образом:

1. Напрямую позиции круговых осей A , B , C
2. Угол Эйлера (в градусах) через $A2$, $B2$, $C2$
3. Угол RPY (в градусах) через $A2$, $B2$, $C2$
4. Вектор направления через $A3$, $B3$, $C3$ (угол поворота с помощью $THETA=<значение>$)

Если ORIROTR или ORIROTТ активны, то угол поворота может быть запрограммирован и напрямую с THETA.

Вращение может быть запрограммировано и одно в кадре, изменения ориентации при этом не происходит. При этом ORIROTR и ORIROTТ не имеют значения. В этом случае угол поворота всегда интерпретируется относительно абсолютного направления (ORIROTA).

N... ORIROTA	Определить интерполяцию вектора вращения
N... ORIROTR	
N... ORIROTТ	
N... ORIROTC	
N... A3= B3= C3= THETA=<значение>	Определить вращение вектора ориентации
N... PO[THET]=(d ₂ , d ₃ , d ₄ , d ₅)	Интерполяция угла поворота с полиномом 5-го порядка

7.4 Вращения ориентации инструмента (ORIROTA, ORIROT, ORIROT, ORIROT, THETA)

Значение

ORIROTA:	Угол поворота к заданному абсолютно направлению вращения
ORIROT:	Угол поворота относительно плоскости между стартовой и конечной ориентацией
ORIROT:	Угол поворота как тангенциальный вектор вращения к изменению ориентации
ORIROT:	Угол поворота как тангенциальный вектор вращения к касательной к траектории
THETA:	Вращение вектора ориентации
THETA=<значение>:	Угол поворота в градусах, достигаемый в конце кадра
THETA=Θ _e :	Угол поворота с конечным углом Θ _e вектора вращения
THETA=AC (...):	Покадровое переключение на абсолютное указание размера
THETA=AC (...):	Покадровое переключение на указание составного размера
Θ _e :	Конечный угол вектора вращения как абсолютно с G90, так к относительно с G91 (указание составного размера) активен
PO[THT]=(...):	Полином для угла поворота

Пример: Вращения ориентаций

Программный код	Комментарий
N10 TRAORI	; Активировать трансформацию ориентации
N20 G1 X0 Y0 Z0 F5000	; Ориентация инструмента
N30 A3=0 B3=0 C3=1 THETA=0	; в направлении Z с углом поворота 0
N40 A3=1 B3=0 C3=0 THETA=90	; в направлении X и поворот на 90 градусов
N50 A3=0 B3=1 C3=0 PO[THT]=(180, 90)	; Ориентация
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; в направлении Y и поворот на 180 градусов
N70 ORIROT	; остается постоянной и поворот на 90 градусов
N80 A3=1 B3=0 C3=0 THETA=30	; Угол поворота относительно изменения ориентации
	; Вектор вращения под углом в 30 градусов к плоскости X-Y

При интерполяции кадра N40 выполняется линейная интерполяция от начального значения 0 градусов до конечного значения 90 градусов. В кадре N50 угол поворота изменяется с 90 градусов до 180 градусов по параболе $\theta(u) = +90u^2$. В N60 также может быть выполнено вращение, изменения ориентации при этом не происходит.

На N80 ориентация инструмента поворачивается из направления Y в направление X. При этом изменение ориентации лежит в плоскости X-Y, а вектор вращения образует угол в 30 градусов к этой плоскости.

Дополнительная информация

ORIROTA

Угол поворота THETA интерпретируется относительно определенного абсолютно направления в пространстве. Базовое направление вращения выполняется через машинные данные.

ORIROTR

Угол поворота **THETA** интерпретируется относительно плоскости, образуемой начальной и конечной ориентацией.

ORIROTТ

Угол поворота **THETA** интерпретируется относительно изменения ориентации. Для **THETA=0** вектор вращения интерполируется тангенциально изменению ориентации и отличается от **ORIROTR** только в том случае, если для ориентации был запрограммирован как минимум один полином для "угла поворота **PSI**". Тем самым получается изменение ориентации, которое выполняется не в плоскости. Тогда через дополнительно запрограммированный угол поворота **THETA** возможна, к примеру, такая интерполяция вектора вращения, при которой он всегда образует определенное значение для изменения ориентации.

ORIROTC

Интерполяция вектора вращения осуществляется относительно касательной к траектории с программируемым через угол **THETA** смещением. При этом для угла смещения можно запрограммировать и полином $PO[THET] = (c2, c3, c4, c5)$ макс. 5-го порядка.

7.5 Ориентации относительно траектории

7.5.1 Типы ориентаций относительно траектории

С помощью этой расширенной функции относительная ориентация достигается не только на конце кадра, а по всему ходу траектории. Достигнутая в предшествующем кадре ориентация через большую круговую интерполяцию переводится в запрограммированную конечную ориентацию. В принципе, существует две возможности программирования желаемой ориентации относительно траектории:

1. Выполняется интерполяция ориентации инструмента и вращения инструмента посредством ORIPATH, ORPATHTS относительно траектории.
2. Программирование и интерполяция вектора ориентации выполняются как и прежде. С ORIOTC устанавливается вращение вектора ориентации относительно касательной к траектории.

Синтаксис

Тип интерполяции ориентации и вращение инструмента программируются с:

N... ORIPATH	Ориентации относительно траектории
N... ORIPATHS	Ориентация относительно траектории со сглаживанием характеристики ориентации
N... ORIOTC	Интерполяция вектора вращения относительно траектории

Вызванный углом в ходе траектории излом ориентации может быть сглажен с помощью ORIPATHS. Направление и длина хода движения отвода программируется через вектор с компонентами A8=X, B8=Y, C8=Z.

С помощью ORIPATH/ORIPATHS для всего хода траектории могут быть запрограммированы могут быть запрограммированы различные отношения к касательной к траектории через три угла

- LEAD= указание прямого угла относительно траектории и поверхности
- TILT= указание бокового угла относительно траектории и поверхности
- THETA= угол поворота

. Для угла поворота `THETA` с помощью `PO [THT] = ( . . . )` дополнительно могут быть запрограммированы полиномы макс. 5-го порядка.

Примечание**Изготовитель станка**

Следовать указаниям изготовителя станка. Через конфигурируемые машинные и установочные данные для типа ориентации относительно траектории могут быть осуществлены дополнительные настройки. Дополнительные пояснения см.

Литература:

/FB3/ Описание функций "Специальные функции"; 3- до 5-осевая трансформация (F2), глава "Ориентация"

Объяснение

Возможны различные настройки интерполяции углов `LEAD` и `TILT` через машинные данные:

- Запрограммированное с `LEAD` и `TILT` отношение ориентации инструмента действует во всем кадре.
- Прямой угол `LEAD`: вращение вокруг направления вертикально к касательной и вектору нормали
`TILT`: вращение ориентации вокруг вектора нормали.
- Прямой угол `LEAD`: вращение вокруг направления вертикально к касательной и вектору нормали Боковой угол
`TILT`: вращение ориентации вокруг направления касательной к траектории.
- Угол поворота `THETA`: вращение инструмента вокруг себя самого с дополнительной третьей круговой осью в качестве оси ориентации при 6-осевой трансформации.

Примечание

Ориентация относительно траектории в комбинации с `OSC`, `OSS`, `OSSE`, `OSD`, `OST` запрещена.

Интерполяция ориентации относительно траектории `ORIPATH` или `ORIPATHS` и `ORIOTC` не может быть запрограммирована в сочетании со сглаживанием характеристики ориентации с одной из G-команд группы 34. Для этого должна быть активна `OSOF`.

7.5.2 Вращение ориентации инструмента относительно траектории (ORIPATH, ORIPATHS, угол поворота)

При 6-осевой трансформации для произвольной ориентации инструмента в пространстве с помощью третьей круговой оси возможно вращение инструмента вокруг себя самого. При вращении ориентации инструмента относительно траектории с ORIPATH или ORIPATHS можно запрограммировать дополнительное вращение через угол поворота THETA. В качестве альтернативы углы LEAD и TILT могут быть запрограммированы через вектор, лежащий в плоскости вертикально к направлению инструмента.

Изготовитель станка

Следовать указаниям изготовителя станка. Через машинные данные возможны различные установки интерполяции углов LEAD и TILT.

Синтаксис

Вращение ориентации инструмента и инструмента

Тип ориентации инструмента относительно траектории активируется с ORIPATH или ORIPATHS.

N... ORIPATH

Активировать тип ориентации относительно траектории

N... ORIPATHS

Активировать тип ориентации относительно траектории со сглаживанием характеристики ориентации

Активация трех возможных углов с поворотным действием:

N... LEAD=

Угол для запрограммированной ориентации относительно вектора нормали плоскости

N... TILT=

Угол для запрограммированной ориентации в плоскости вертикально к касательной к траектории относительно вектора нормали плоскости

N... THETA=

Угол поворота относительно изменения ориентации вокруг направления инструмента третьей круговой оси

Значения угла на конце кадра программируются с LEAD=значение, TILT=значение или THETA=значение. В дополнение к постоянным углам для всех трех углов могут быть запрограммированы полиномы макс. 5-го порядка.

N... PO[PHI]=(a2, a3, a4, a5)

Полином для угла предварения LEAD

N... PO[PSI]=(b2, b3, b4, b5)

Полином для бокового угла TILT

N... PO[THT]=(d2, d3, d4, d5)

Полином для угла поворота THETA

При программировании более высокие коэффициенты полиномов, равные нулю, могут быть опущены. Пример PO[PHI]=a2 дает параболу для угла предварения LEAD.

Значение

Ориентация инструмента относительно траектории

ORIPATH:	Ориентация инструмента относительно траектории
ORIPATHS :	Ориентация инструмента относительно траектории, изгиб в характеристике ориентации сглаживается
LEAD:	Угол относительно вектора нормали плоскости, в образованной касательной к траектории и вектором нормали плоскости
TILT:	Вращение ориентации вокруг направления Z или вращение вокруг касательной к траектории
THETA:	Вращение вокруг направления инструмента в Z
PO[PHI]:	Полином ориентации для угла предварения LEAD
PO[PSI]:	Полином ориентации для бокового угла TILT
PO[THT]:	Полином ориентации для угла поворота THETA

Примечание

Угол поворота THETA

Для вращения инструмента с помощью третьей круговой оси в качестве оси ориентации вокруг себя самого необходима 6-осевая трансформация.

7.5.3 Интерполяция вращения инструмента относительно траектории (ORIOTC, THETA)

Интерполяция с векторами вращения

К запрограммированному с ORIOTC вращению инструмента относительно касательной к траектории возможна интерполяция вектора вращения и с программируемым через угол поворота THETA смещением. При этом для угла смещения с PO[THT] может быть запрограммирован полином макс. 5-го порядка.

Синтаксис

N... ORIOTC	Установка вращения инструмента относительно касательной к траектории
N... A3= B3= C3= THETA=значение	Определить вращение вектора ориентации
N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)	Интерполяция угла смещения с полиномом макс. 5-го порядка

Вращение может быть запрограммировано и одно в кадре, изменения ориентации при этом не происходит.

Значение

Интерполяция вращения инструмента относительно траектории при 6-осевой трансформации

ORIROTС:	Установить тангенциальный вектор вращения к касательной к траектории
THETA=значение:	Угол поворота в градусах, достигаемый в конце кадра
THETA=Θ _е :	Угол поворота с конечным углом Θ _е вектора вращения
THETA=AC (...):	Покадровое переключение на абсолютное указание размера
THETA=IC (...):	Покадровое переключение на указание составного размера
PO[THT]=(с2, с3, с4, с5):	Интерполяция угла смещения с полиномом 5-го порядка

Примечание**Интерполяция вектора вращения ORIROTС**

Если против направления ориентации инструмента должно быть установлено и вращение инструмента относительно касательной к траектории, то это возможно только при 6-осевой трансформации.

При активной ORIROTС

Вектор вращения ORIROTA не может быть запрограммирован. В случае программирования выводится ОШИБКА 14128 "Абсолютное программирование вращения инструмента при активной ORIROTС".

Направление ориентации инструмента при 3- до 5-осевой трансформации

Направление ориентации инструмента может быть запрограммировано как обычно при 3- до 5-осевой трансформации через угол Эйлера или угол RPY или векторы направления. Изменения ориентации инструмента в пространстве также возможны через программирование большой круговой интерполяции ORIVECT, линейной интерполяции осей ориентации ORIAXES, всех интерполяций на боковой поверхности конуса ORICONxx, а также интерполяции дополнительно к пространственной кривой с двумя контактными точками инструмента ORICURVE.

G....:	Указание типа движения круговых осей
X, Y, Z:	Указание линейных осей
ORIAXES:	Линейная интерполяция осей станка или осей ориентации
ORIVECT:	Большая круговая интерполяция (идентично ORIPLANE)
ORIMKS:	Вращение в системе координат станка
ORIWKs:	Вращение в системе координат детали
	Описание см. в гл. Вращения ориентации инструмента
A= B= C=:	Программирование позиции оси станка
ORIEULER:	Программирование ориентации через угол Эйлера
ORIRPY:	Программирование ориентации через угол RPY
A2= B2= C2=:	Программирование угла виртуальных осей

ORIVIRT1: ORIVIRT2:	Программирование ориентации через виртуальные оси ориентации (Определение 1), установка по MD \$MC_ORIAX_TURN_TAB_1 (Определение 2), установка по MD \$MC_ORIAX_TURN_TAB_2
A3= B3= C3=:	Программирование вектора направления оси направления
ORIPLANE:	Интерполяция в плоскости (большая круговая интерполяция)
ORICONCW:	Интерполяция на боковой поверхности конуса по часовой стрелке
ORICONCCW:	Интерполяция на боковой поверхности конуса против часовой стрелки
ORICONTO:	Интерполяция на боковой поверхности конуса, тангенциальный переход
A6= B6= C6=:	Программирование оси вращения конуса (нормированный вектор)
NUT=угол	Апертурный угол конуса в градусах
NUT=+179	Угол перемещения меньше или равен 180 градусам
NUT=-181	Угол перемещения больше или равен 180 градусам
ORICONIO:	Интерполяция на боковой поверхности конуса
A7= B7= C7=:	Промежуточная ориентация (программирование как нормированный вектор)
ORICURVE XH YH ZH, к примеру, с полиномом PO[XH]=(xe, x2, x3, x4, x5)	Интерполяция ориентации с задачей движения двух контактных точек инструмента. Кроме соответствующих конечных точек, возможно программирование дополнительных полиномов пространственных кривых.

Примечание

Если интерполяция ориентации инструмента с активной ORIAXES осуществляется через оси ориентации, то установка угла поворота относительно траектории выполняется только на конце кадра.

7.5.4**Сглаживание характеристики ориентации (ORIPATHS A8=, B8=, C8=)**

При изменениях ориентации на контуре с постоянным ускорением, нежелательными являются прерывания движений по траектории, в особенности на углу контура. Возникающий из-за этого излом характеристики ориентации может быть сглажен через вставку собственного промежуточного кадра. Изменение ориентации выполняется с постоянным ускорением тогда, когда при переориентации активна и ORIPATHS. На этой фазе может быть выполнено движение отвода инструмента.

Изготовитель станка

Следовать указаниям изготовителя станка касательно возможно predetermined машинных данных и установочных данных, с помощью которых активируется эта функция.

Через машинные данные может быть установлено, как интерпретируется вектор отвода:

1. В системе координат инструмента координата Z определяется через направление инструмента.
2. В системе координат детали координата Z определяется через активную плоскость.

Дополнительные пояснения к функции "Ориентация относительно траектории" см.

Литература:

Описание функций - Специальные функции; Многоосевые трансформации (F2)

Синтаксис

Для постоянных ориентаций инструмента относительно всей траектории на углу контура необходимы дополнительные данные программирования. Направление и длина хода этого движения программируется через вектор с компонентами A8=X, B8=Y, C8=Z:

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Значение

ORIPATHS:	Ориентация инструмента относительно траектории, излом в характеристике ориентации сглаживается.
A8= B8= C8=:	Компоненты вектора для направления и длины хода
X, Y, Z:	Движение отвода в направлении инструмента

Примечание**Программирование вектора направления A8, B8, C8**

Если длина этого вектора равна нулю, то движение отвода не выполняется.

ORIPATHS

Относящаяся к траектории ориентация инструмента активируется с ORIPATHS. В ином случае ориентация посредством линейной большой круговой интерполяции переводится от стартовой к конечной ориентации.

7.6 Сжатие ориентации (COMPON, COMPCURV, COMPCAD, COMPSURF)

Программы ЧПУ, в которых активна трансформация ориентации (TRAORI) и запрограммированы ориентации инструмента (любого типа), могут быть сжаты при соблюдении заданных допусков.

Программирование

Ориентация инструмента

Если активна трансформация ориентации (TRAORI), программирование ориентации инструмента у 5-осевых станков возможно следующим образом (независимо от кинематики):

- Программирование **вектора** направления через:
A3=<...> B3=<...> C3=<...>
- Программирование **угла** Эйлера или **угла** RPY через:
A2=<...> B2=<...> C2=<...>

Вращения инструмента

Для **6-осевых** станков, дополнительно к ориентации инструмента, можно запрограммировать и вращение инструмента.

Программирование угла поворота осуществляется с:

THETA=<...>

См. " Вращения ориентации инструмента (Страница 368) ".

Примечание

Кадры УП, в которых дополнительно запрограммировано вращение, могут сжиматься только тогда, когда угол поворота изменяется **линейно**. Т. е. для угла поворота не может быть запрограммировано полинома с PO[THET]=(...).

Общая форма сжимаемого кадра УП

Поэтому общая форма сжимаемого кадра ЧПУ может выглядеть следующим образом:

N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>

или

N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>

Примечание

Значения позиция могут указываться напрямую (например, X90) или косвенно через присваивания параметров (например, X=R1*(R2+R3)).

Программирование ориентации инструмента через позиции круговых осей

Ориентация инструмента может быть указана и через позиции круговых осей, например, в форме:

N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>

В этом случае сжатие выполняется двумя различными способами, в зависимости от того, выполняется ли большая круговая интерполяция или нет. Если большая круговая интерполяция не осуществляется, то сжатое изменение ориентации отображается как обычно через осевые полиномы для круговых осей.

Точность контура

В зависимости от установленного режима сжатия (MD20482 \$MC_COMPRESSOR_MODE), для гео-осей и осей ориентации при сжатии действует либо сконфигурированные, спец. для осей допуски (MD33100 \$MA_COMPRESS_POS_TOL), либо следующие, настраиваемые через установочные данные спец. для канала допуски:

SD42475 \$SC_COMPRESS_CONTUR_TOL (макс. погрешность контура)

SD42476 \$SC_COMPRESS_ORI_TOL (макс. угловая погрешность для ориентации инструмента)

SD42477 \$SC_COMPRESS_ORI_ROT_TOL (макс. угловая погрешность для угла поворота инструмента) (доступно только для 6-осевых станков)

Литература:

Описание функций "Основные функции"; 3- до 5-осевая трансформация (F2), глава: "Сжатие ориентации"

Активация / деактивация

Функции компрессора включаются через модальные команды G COMPON, COMPCURV, COMPCAD или COMPSURF.

Завершение функции компрессора осуществляется с помощью COMPOF.

См. " Сжатие кадров ЧПУ (COMPON, COMPCURV, COMPCAD) (Страница 274) ".

Примечание

Движение ориентации сжимается только при активной большой круговой интерполяции (т. е. изменение ориентации инструмента осуществляется в плоскости, образуемой стартовой и конечной ориентацией).

Большая круговая интерполяция выполняется при следующих условиях:

- MD21104 \$MC_ORI_IPO_WITH_G_CODE = 0,
ORIWKS активна и
ориентация запрограммирована посредством векторов (с A3, B3, C3 или A2, B2, C2).
- MD21104 \$MC_ORI_IPO_WITH_G_CODE = 1 и
ORIVECT или ORIPLANE активна.
Ориентация инструмента может быть запрограммирована либо как вектор направления, либо с позициями круговых осей. Если активна одна из команд ORICONxx или ORICURVE, или если запрограммированы полиномы для углов ориентации (PO[PHI] и PO[PSI]), то большая круговая интерполяция не выполняется.

Пример

В примере программы ниже сжимается окружность, приближенная через полигон. Ориентация инструмента при этом перемещается синхронно на боковой поверхности конуса. Хотя ход последовательных запрограммированных изменений ориентации является дискретным, компрессор создает ровную характеристику ориентации.

Программирование	Комментарий
DEF INT ANZAHL=60 DEF REAL RADIUS=20 DEF INT COUNTER DEF REAL WINKEL N10 G1 X0 Y0 F5000 G64 \$SC_COMPRESS_CONTUR_TOL=0.05 \$SC_COMPRESS_ORI_TOL=5 TRAORI COMPCURV N100 X0 Y0 A3=0 B3=-1 C3=1 N110 FOR COUNTER=0 TO ANZAHL N120 WINKEL=360*COUNTER/ANZAHL N130 X=RADIUS*cos(WINKEL) Y=RADIUS*sin(WINKEL) A3=sin(WINKEL) B3=-cos(WINKEL) C3=1 N140 ENDFOR	 ; Макс. погрешность контура = 0.05 мм ; Макс. погрешность ориентации = 5 градусов ; Проходится круг, образуемый из полигонов. Ориентация при этом движется на конусе вокруг оси Z с апертурным углом в 45 градусов.

7.7 Включение/выключение сглаживания характеристики ориентации (ORISON, ORISOF)

"Сглаживание характеристики ориентации" включается/выключается в программе обработки детали G-командами группы 61. Команды действуют модально.

Условия

- Система с 5/6-осевой трансформацией.
- Функция компрессора COMPCAD активна.

Синтаксис

```
ORISON
...
ORISOF
```

Объяснение

ORISON:	Включение сглаживания характеристики ориентации
ORISOF:	Выключение сглаживания характеристики ориентации

Пример

Программный код	Комментарий
...	
TRAORI ()	; Включение трансформации ориентации.
COMPCAD	; Включение функции компрессора COMPCAD
ORISON	; Включение сглаживания ориентации.
\$SC_ORISON_TOL=1.0	; Макс. угловое отклонение ориентации инструмента = 1,0 градуса.
G91	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
...	
ORISOF	; Выключение сглаживания ориентации.
...	

7.7 Включение/выключение сглаживания характеристики ориентации (ORISON, ORISOF)

Ориентация поворачивается на 90 градусов в плоскости XZ с -45 до +45 градусов. Из-за сглаживания характеристики ориентации, ориентация более не достигает макс. угловых значений в -45 или +45 градусов.

7.8 Кинематическая трансформация

7.8.1 Включение трансформации с торцевой стороны (TRANSMIT)

Трансформация с торцевой стороны (TRANSMIT) включается в программе обработки детали или синхронном действии через оператор TRANSMIT.

Синтаксис

```
TRANSMIT
TRANSMIT (<n>)
```

Объяснение

TRANSMIT:	Включить TRANSMIT с первым блоком данных TRANSMIT
TRANSMIT (n):	Включить TRANSMIT с <n>-ным блоком данных TRANSMIT

Примечание

Активная в канале трансформация TRANSMIT выключается через:

- выключение трансформации: TRAFOOF
- включение другой трансформации: например, TRACYL, TRAANG, TRAORI

7.8.2 Включение трансформации боковой поверхности цилиндра (TRACYL)

Трансформация боковой поверхности цилиндра (TRACYL) включается в программе обработки детали или синхронном действии через оператор TRACYL.

Синтаксис

```
TRACYL (<d>)
TRACYL (<d>, <n>)
TRACYL (<d>, <n>, <k>)
```

Объяснение

TRACYL (<d>):	Включить TRACYL с первым блоком данных TRACYL и рабочим диаметром <d>
TRACYL (<d>, <n>):	Включить TRACYL с <n>-ным блоком данных TRACYL и рабочим диаметром <d>
<d>:	Базовый или рабочий диаметр Значение может быть больше 1.

<n>:	TRACYL-номер блока данных (опция)	
	Диапазон значений:	1, 2
<k>:	Параметр <k> релевантен только для типа трансформации 514	
	k = 0:	без коррекции стенки паза
	k = 1:	с коррекцией стенки паза
	Если параметр не указывается, действует спараметрированная первичная установка: \$MC_TRACYL_DEFAULT_MODE_<n> с <n> = TRACYL-номер блока данных	

Примечание

Активная в канале трансформация TRACYL выключается через:

- выключение трансформации: TRAFOOF
- включение другой трансформации: например, TRAANG, TRANSMIT, TRAORI

Пример

Программный код	Комментарий
...	
N40 TRACYL (40.)	; Включить TRACYL с первым блоком данных TRACYL и рабочим диаметром 40 мм.
...	

Дополнительная информация**Структура программы**

Программа обработки детали для фрезерования паза с помощью трансформации TRACYL 513 (TRACYL с коррекцией стенки паза), как правило, состоит из следующих шагов:

1. Выбрать инструмент.
2. Выбрать TRACYL.
3. Выбрать подходящее смещение координат (FRAME).
4. Выполнить позиционирование.
5. Запрограммировать OFFN.
6. Выбрать КРИ.
7. Кадр подвода (ввод КРИ и подвод к стенке паза).
8. Контур линии центров паза.
9. Отключить КРИ.
10. Кадр отвода (вывод КРИ и отвод от стенки паза).

11. Выполнить позиционирование.

12. TRAFOOF.

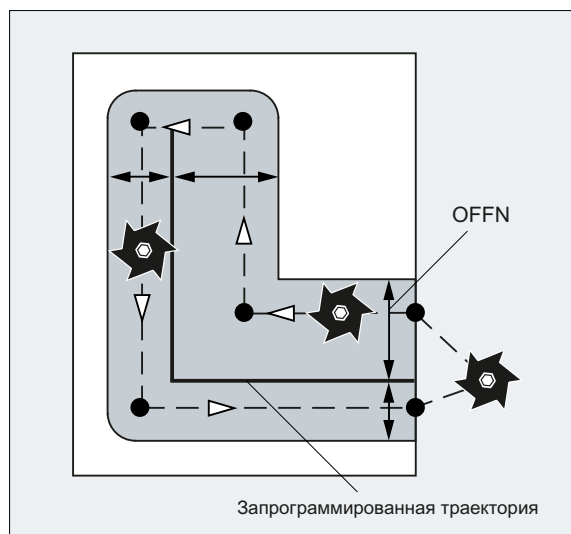
13. Снова выбрать первоначальное смещение координат (FRAME).

Смещение контура (OFFN)

Для фрезерования пазов с трансформацией TRACYL 513, в программе обработки детали программируется линия центров паза и через адрес OFFN **половина ширины паза**.

Во избежание повреждения стенки паза, OFFN активируется только при выбранной коррекции на радиус инструмента.

Изменение OFFN внутри программы обработки детали возможно. Тем самым линия центра паза может быть смещена из центра:



Примечание

OFFN должно быть не меньше радиуса инструмента, чтобы исключить повреждение противоположной стенки паза.

Примечание

OFFN с TRACYL действует по иному, чем без TRACYL. Т.к. OFFN и без TRACYL учитывается при активной КРИ, то после TRAFOOF необходимо снова установить OFFN на ноль.

ВНИМАНИЕ**Действие OFFN зависит от типа трансформации**

При трансформации TRACYL 513 (TRACYL с коррекцией стенки паза) для OFFN программируется половина ширины паза.

А при трансформации TRACYL 512 (TRACYL без коррекции стенки паза) значение из OFFN действует как припуск к КРИ.

Коррекция на радиус инструмента (КРИ)

При трансформации TRACYL 513 КРИ учитывается не относительно стенки паза, а к запрограммированному центру паза. Чтобы инструмент проходил слева от стенки паза, вместо G41 нужно запрограммировать оператор G42 или указать значение OFFN с отрицательным знаком.

Диаметр инструмента

С TRACYL и инструментом, диаметр которого меньше, чем ширина паза, создается геометрия стенки паза, отличная от таковой, создаваемой с помощью инструмента, диаметр которого равен ширине паза. Для улучшения точности рекомендуется выбирать диаметр инструмента лишь ненамного меньше, чем ширина паза.

Использование осей**Примечание**

Следующие оси не могут использоваться в качестве позиционирующей или качающейся оси:

- Гео-ось в направлении обвода боковой поверхности цилиндра (ось Y)
- Дополнительная линейная ось для коррекции стенки паза (ось Z)

7.8.3 Включение косоугольной трансформации с программируемым углом (TRAANG)

Косоугольная трансформация с программируемым углом включается оператором TRAANG в программе обработки детали или синхронном действии.

Синтаксис

```
TRAANG
TRAANG ( )
TRAANG ( , <n> )
TRAANG ( <α> )
TRAANG ( <α> , <n> )
```

Объяснение

TRAANG: TRAANG () :	Включить TRAANG с первым блоком данных TRAANG и последним действительным углом <α>	
TRAANG (, <n>) :	Включить TRAANG с <n>-ным блоком данных TRAANG и последним действительным углом <α>	
TRAANG (<α>) :	Включить TRAANG с первым блоком данных TRAANG и углом <α>	
TRAANG (<α> , <n>) :	Включить TRAANG с <n>-ным блоком данных TRAANG и углом <α>	
<α>:	Угол наклонной оси (опция)	
	Диапазон значений:	-90° < α < + 90°
	Если угол не указывается, действует первичная установка из машинных данных: MD2xxxx \$MC_TRAANG_ANGLE_<n>	
<n>:	TRAANG-номер блока данных (опция)	
	Диапазон значений:	1, 2

Примечание

Активная в канале косоугольная трансформация TRAANG выключается через:

- выключение трансформации: TRAFOOF
- включение другой трансформации: например, TRACYL, TRANSMIT, TRAORI

Пример

Программный код	Комментарий
N20 TRAANG(45)	; Включить TRAANG с первым блоком данных TRAANG и углом 45°.

7.8.4 Наклонное врезание на шлифовальных станках (G5, G7)

G-команды G7 и G5 служат для упрощения программирования при наклонном врезном шлифовании на шлифовальных станках с трансформацией «Наклонная ось (TRAANG)», при этом при врезном шлифовании перемещается только наклонная ось.

При этом программируется только требуемая конечная позиция движения врезания в X и Z. Соответствующая стартовая позиция вычисляется ЧПУ для G7, исходя из актуальной позиции оси X, запрограммированной конечной позиции и угла α наклонной оси и выполняется подвод.

Стартовая позиция получается из точки пересечения обеих прямых:

- Прямая, параллельная ось Z, на расстоянии текущей позиции оси X
- Прямая параллельно наклонной оси через запрограммированную конечную позицию

При следующей G5 наклонная ось движется на запрограммированную конечную позицию.

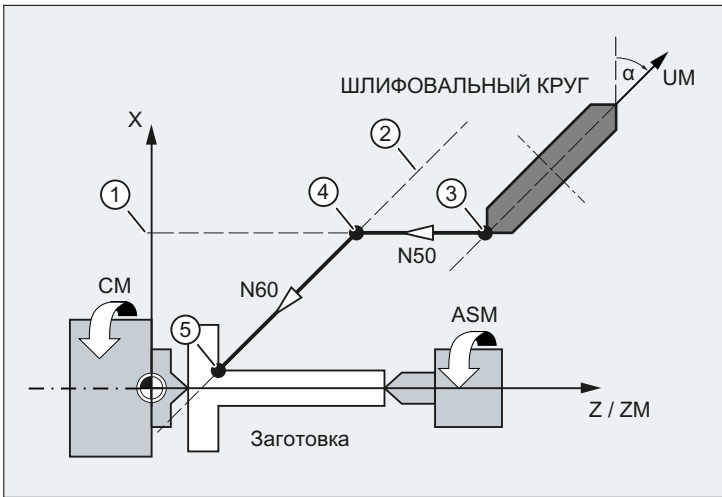
Синтаксис

```
G7 <Endpos_X> <Endpos_Z>
G5 <Endpos_X>
```

Объяснение

G7:	Вычисление стартовой позиции и подвод в нее для наклонного врезания
G5:	Перемещение наклонной оси на запрограммированную конечную позицию
<Endpos_X>:	Конечная позиция оси X
<Endpos_Z>:	Конечная позиция оси Z

Пример



- ① Параллельно оси Z, на расстоянии текущей позиции оси X
- ② Параллельно наклонной оси через запрограммированную конечную позицию
- ③ Исходное положение
- ④ Врезание: Стартовая позиция
- ⑤ Врезание: Конечная позиция
- X Гео-ось
- Z Гео-ось
- ZM Ось станка
- UM Ось станка

Программный код	Комментарий
N... G18	; Выбор плоскости XZ.
N40 TRAANG (45.0)	; Включить трансформацию TRAANG, угол = 45°.
N50 G7 X40 Z70 F4000	; Вычисление стартовой позиции и подвод в нее.
N60 G5 X40 F100	; Перемещение наклонной оси на конечную позицию.
N70 ...	

7.9 Включить связанную трансформацию (TRACON)

Связанная трансформация в программе обработки детали или синхронном действии включается через оператор TRACON.

Синтаксис

```
TRACON (<трансф_№>,<пар_1>,<пар_2>,...)
...
TRAFOOF
```

Объяснение

TRACON:	Включить связанную трансформацию Активированная прежде другая трансформация не явно выключается через TRACON () .			
<трансф_№>:	Номер связанной трансформации			
	Тип:	INT		
	Диапазон значений:	0 ... 2		
	Значение:	0, 1	первая/единственная связанная трансформация	
		2	вторая связанная трансформация	
		не указано	равнозначно 0 или 1	
		Примечание: Значения, отличные от 0, 1, 2, вызывают сигнализацию сбоя.		
<пар_1>,<пар_2,... ..	Параметры для связанных друг с другом трансформаций (например, угол наклонной оси) При не установленных параметрах активируются предустановки или последнее использовавшиеся параметры. Посредством запятой устанавливается ожидаемая последовательность обработки указанных параметров, если для предстоящих параметров должны действовать предустановки. В частности при указании минимум одного параметра перед ним должна стоять запятая, даже если указание <трансф_№> не обязательно, например, TRACON (, 3.7) . В случае TRACON с Transmit или TRAORI (5-осевая обработка) второй параметр Par_2 не действует как угол наклонной оси. В случае TRACON с TRACYL (обработка боковой поверхности) Par используется для единого диаметра.			
TRAFOOF:	Выключение последней включенной (связанной) трансформации			

Пример

Программный код	Комментарий
...	

7.9 Включить связанную трансформацию (TRACON)

Программный код	Комментарий
N230 TRACON(1,45.)	; Включить первую связанную трансформацию.
...	; Активная прежде трансформация отключается автоматически.
...	; Параметр для наклонной оси равен 45°.
N330 TRACON(2,40.)	; Включить вторую связанную трансформацию.
...	; Параметр для наклонной оси равен 40°.
...	
N380 TRAFOOF	; Выключить вторую связанную трансформацию.
...	

7.10 Движение “от точки к точке” в декартовой системе координат

7.10.1 Включение / выключение движения “от точки к точке” в декартовой системе координат (PTP, RTPG0, RTPWOC, CP)

Движение “от точки к точке” в декартовой системе координат включается / выключается в программе ЧПУ командами группы G 49.

Команды действуют модально. По умолчанию задано перемещение по траектории в декартовой системе координат (CP).

В отличие от коммуникационного процессора при активном движении от точки к точке выполняется только трансформация конечной точки в декартовой системе координат, и оси станка перемещаются синхронно.

Чтобы конечную точку в декартовой системе координат можно было пересчитать в значения осей станка, дополнительно к указанию положения и угла необходима информация о маркировке положений осей. Эти данные вводятся при помощи настраиваемых адресов STAT (Страница 392) и TU (Страница 396).

Условие

Трансформация TRAORI, TRANSMIT, RCTRA или ROBX активна.

Синтаксис

```
PTP / RTPG0 / RTPWOC
...
CP
```

Объяснение

PTP:	Включение позиционного движения PTP Переход в запрограммированную позицию в декартовой системе координат в кадрах G0 и G1 осуществляется при помощи движения синхронной оси.
RTPG0:	Включение позиционного движения RTPG0 Переход в запрограммированную позицию в декартовой системе координат осуществляется при помощи движения синхронной оси только в кадрах G0. В кадрах G1 выполняется переключение на движение по траектории CP.
RTPWOC:	Включение позиционного движения (“от точки к точке”) RTPWOC (возможно только при активной трансформации ориентации) Аналогично PTP, разумеется, без компенсирующих движений, обусловленных движением круговых осей и осей ориентации.
CP:	Выключение позиционного движения и включение движения по траектории CP С помощью CP выполняется движение по траектории в декартовой системе координат.

Примечание**РТРWOC**

В сочетании с трансформацией RCTRA или ROBX использование РТРWOC не имеет смысла!

Примеры

См.:

- Пример 1: Движение “от точки к точке” 6-осевого робота с трансформацией ROBX (Страница 399)
- Пример 2: Движение "от точки к точке" при базовой 5-осевой трансформации (Страница 400)
- Пример 3: РТПG0 и (Страница 400)

7.10.2 Задание положения шарниров (STAT)

Указания положения в декартовых координатах и ориентации инструмента недостаточно для однозначного определения положения станка, так как в одной и той же ориентации инструмента допускается несколько позиций сочленения. В зависимости от того, о какой кинематике идет речь, существует до 8 различных позиций сочленения. Эти различные позиции сочленения являются спец. для трансформаций.

Во избежание неточностей положение сочленений задается в адресе STAT.

Примечание

Управление учитывает запрограммированные значения STAT только при позиционном движении. При движении CP они игнорируются, так как при перемещении с активной трансформацией смена позиции, как правило, невозможна. При перемещении с активной командой CP позиция для конечной точки берется из начальной точки.

Синтаксис

STAT=<значение>

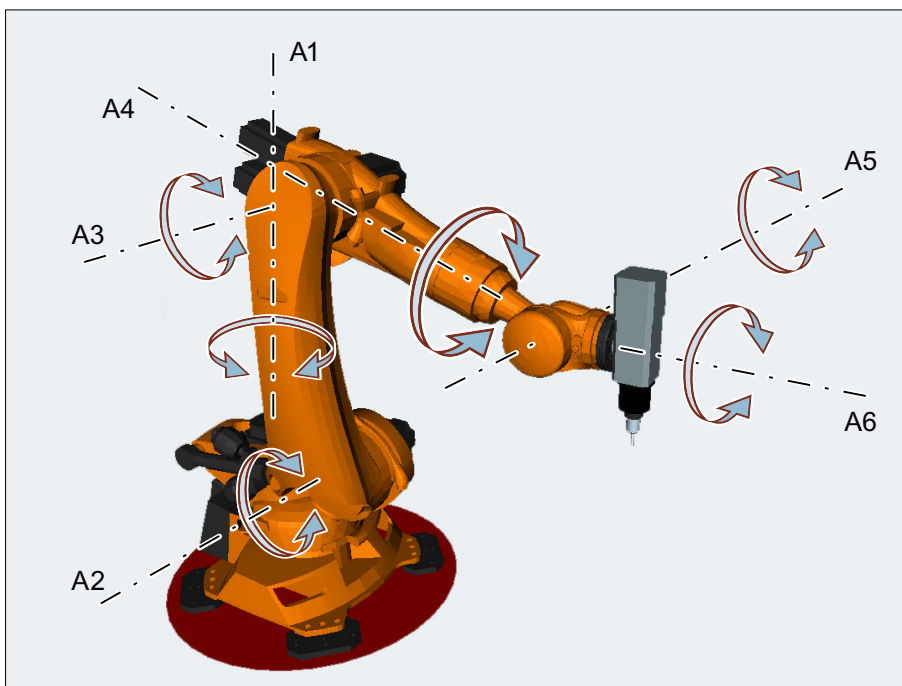
Объяснение

STAT:	Настраиваемый адрес для указания позиций сочленений
<значение>:	Двоичное или десятичное значение Содержит один бит для каждой из возможных позиций. Значение битов определяется соответствующей трансформацией.

Использование STAT необходимо проиллюстрировать на примере 6-осевого робота со сгибающейся рукой с фрезерным шпинделем. Кинематическая трансформация должна

7.10 Движение "от точки к точке" в декартовой системе координат

выполняться посредством трансформации ROBX (условие: Компилируемый цикл "Расширенная роботизированная трансформация RMCC/ROBX " загружен и активен).



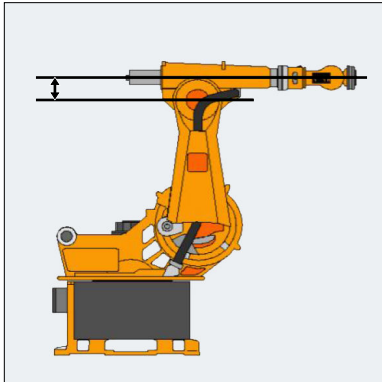
Оси A1, A2 и A3 — главные оси робота со сгибающейся рукой. Посредством главных осей оси A4, A5 и A6, обозначаемые "головные" или "ручные" оси, размещаются в рабочем пространстве. Благодаря дополнительным возможностям движения ручных осей фрезерный шпиндель можно ориентировать в пространстве так, как это необходимо для задач обработки. При этом при одновременной ориентации инструмента возможны различные положения шарнира.

Используемые для обработки положения шарнира программируются (бит 0 ... 2 настраиваемого адреса STAT:

Бит 0	позиция точки пересечения ручных осей (A4, A5, A6)	
	=0	Базовая область (Shoulder Right) Робот находится в базовой области, когда значение X точки пересечения ручных осей положительно в системе координат A1.
	=1	Верхняя область (Shoulder Left) Робот находится в верхней области, когда значение X точки пересечения ручных осей отрицательно в системе координат A1.

Пример: Точка пересечения "ручных" осей находится в базовой области

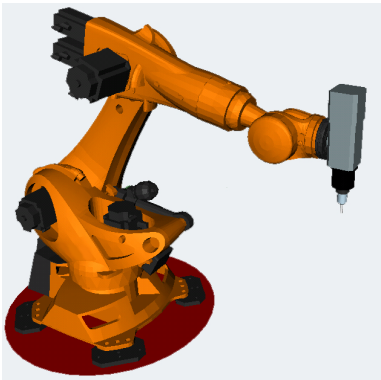
7.10 Движение “от точки к точке” в декартовой системе координат

Бит 1	позиция оси 3	
	Угол, при котором изменяется значение бита 1, зависит от типа робота. Для роботов, оси 3 и 4 которых пересекаются, справедливо:	
	=0	A3 < 0° (Elbow Down)
	=1	A3 ≥ 0° (Elbow Up)
	Примечание: Для роботов со смещением между осями 3 и 4 угол, при котором изменяется значение бита 1, зависит от величины этого смещения.	
		
	Смещение между A3 и A4.	
Бит 2	позиция оси 5	
	=0	A5 ≤ 0° (No Handflip)
	=1	A5 > 0° (Handflip)

Пример программы:

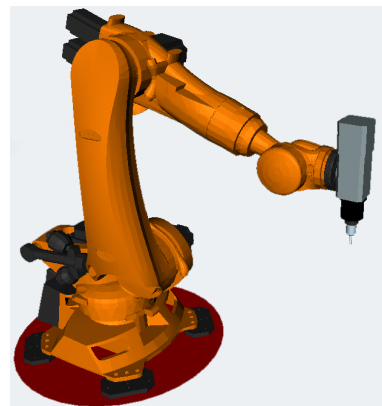
Программный код	Комментарий
...	
N14 T="T8MILLD20" D1	; \$TC_DP3[1,1]=132.95
N16 ORIMKS	
N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 STAT=... F2000	; Значение STAT определяет положения шарнира (см. далее).
...	

STAT=1 ('B001') → Shoulder Left
 → Elbow Down
 → No Handflip



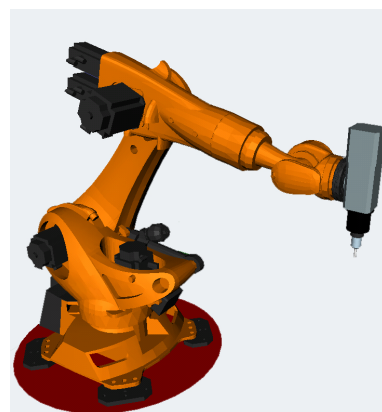
STAT=2 ('B010')

- Shoulder Right
- Elbow Up
- No Handflip



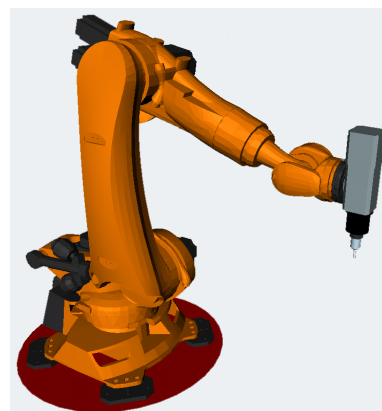
STAT=5 ('B101')

- Shoulder Left
- Elbow Down
- Handflip



STAT=6 ('B110')

- Shoulder Right
- Elbow Up
- Handflip



Transmit

В TRANSMIT используется адрес STAT, позволяющий исключить многозначность относительно полюса.

Если круговая ось должна вращаться на 180° , или контур при CP должен идти через полюс:

Бит 0	Относится только к \$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1 или 2:	
	=0	Круговая ось перемещается на $+180^\circ$ или поворачивается по часовой стрелке.
	=1	Круговая ось перемещается на -180° или поворачивается против часовой стрелки.
Бит 1	Относится только к \$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0:	
	=0	Выполняется перемещение через полюс. Круговая ось не поворачивается.
	=1	Выполняется поворот вокруг полюса. При этом имеет значение бит 0 в STAT.

7.10.3 Знак угла между осями (TU)

Чтобы оси вращения могли поворачиваться на угол между осями, величина которого больше $+180^\circ$ или меньше -180° , без применения специальной стратегии перемещения (например, промежуточная точка), необходимо задать в настраиваемом адресе TU знак угла между осями.

Примечание

Управление учитывает запрограммированные значения TU только при позиционном движении. При движении CP они игнорируются.

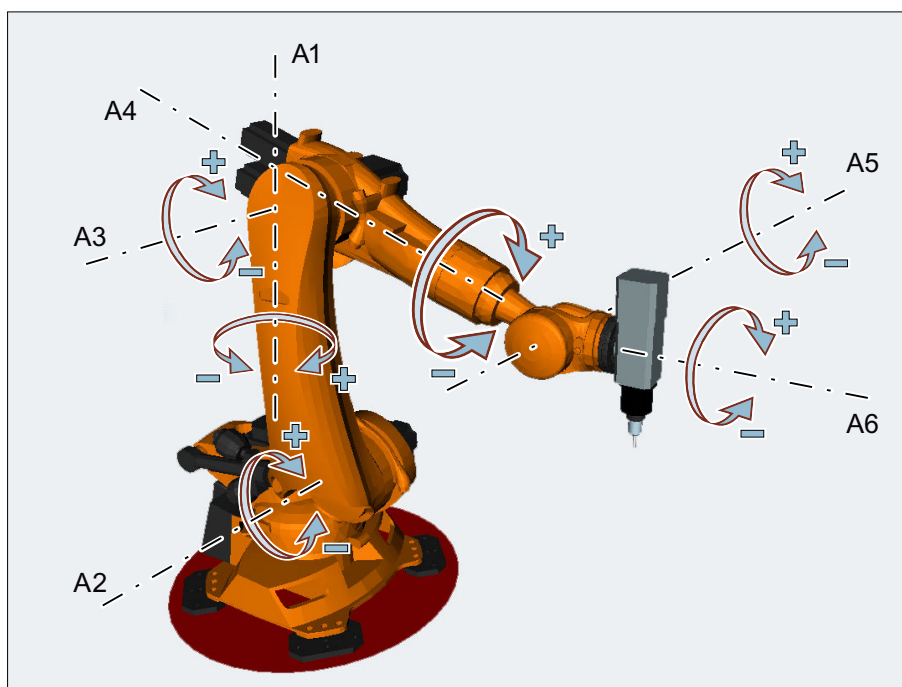
Синтаксис

TU=<значение>

Объяснение

TU:	Настраиваемый адрес для указания знака угла между осями			
<значение>:	Двоичное или десятичное значение			
	Для каждой оси, участвующей в трансформации, имеется бит, указывающий знак угла между осями (θ) и, тем самым, направление перемещения.			
	Бит			
		=0	Знак перед осевым углом: +	Диапазон осевых углов: $0^\circ \leq \theta < 360^\circ$
		=1	Знак перед осевым углом: -	Диапазон осевых углов: $-360^\circ < \theta < 0^\circ$

Пример: 6-осевой робот со сгибающейся рукой



Бит	Объяснение	Значение	Знак перед осевым углом	Осевой угол
Бит 0 ¹⁾	Знак перед осевым углом A1	=0	+	$\geq 0^\circ$
		=1	-	$< 0^\circ$
Бит 1 ¹⁾	Знак перед осевым углом A2	=0	+	$\geq 0^\circ$
		=1	-	$< 0^\circ$
Бит 2 ¹⁾	Знак перед осевым углом A3	=0	+	$\geq 0^\circ$
		=1	-	$< 0^\circ$
Бит 3 ¹⁾	Знак перед осевым углом A4	=0	+	$\geq 0^\circ$
		=1	-	$< 0^\circ$
Бит 4 ¹⁾	Знак перед осевым углом A5	=0	+	$\geq 0^\circ$
		=1	-	$< 0^\circ$
Бит 5 ¹⁾	Знак перед осевым углом A6	=0	+	$\geq 0^\circ$
		=1	-	$< 0^\circ$

¹⁾ Фактические номера битов TU получаются из номеров осей канала осей робота! В примере оси робота (с A1 по A6) — это первые шесть осей в канале, следовательно, используются биты TU 0 ... 5. При другом соотношении осей канала робота номера битов TU осей робота соответствующим образом изменятся (з. В.: оси робота — . это оси канала с 3 по 8, т. е. для осей робота используются биты TU 2 ... 7).

TU=19 (соответствует TU='B010011) будет также обозначать:

Бит	Значение	Осевой угол
0	=1 $\Rightarrow$	$\theta_{A1} < 0^\circ$
1	=1 $\Rightarrow$	$\theta_{A2} < 0^\circ$

2	=0	⇒	$\theta_{A3} \geq 0^\circ$
3	=0	⇒	$\theta_{A4} \geq 0^\circ$
4	=1	⇒	$\theta_{A5} < 0^\circ$
5	=0	⇒	$\theta_{A6} \geq 0^\circ$

Примечание

Для осей с диапазоном перемещения $> \pm 360^\circ$ перемещение всегда выполняется по кратчайшему пути, т.к. позиция оси не может быть однозначно определена через TU-информацию.

Если в позиции TU не программируется, то в зависимости от MD30455

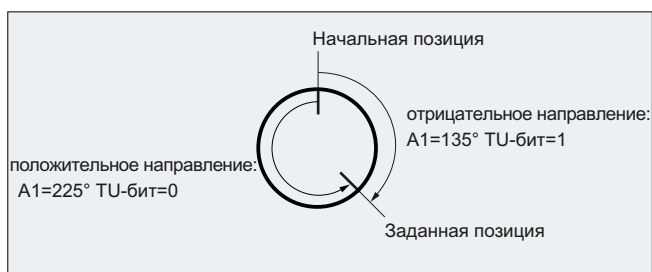
\$MA_MISC_FUNCTION_MASK проходится более короткий или более длинный путь (см. главу Учет предела SW при движении "от точки к точке" (PTP) в Описание функций "Дополнительные функции").

Transmit

При позиционном перемещении при активном TRANSMIT адрес TU не имеет значения!

Пример

Подвод к указанной на следующем рисунке позиции круговой оси возможен в отрицательном или положительном направлении. По адресу A1 программируется положение угла. Однозначное направление перемещения задается только параметром TU.



7.10.4 Пример 1: Движение "от точки к точке" 6-осевого робота с трансформацией ROBX

В следующем примере использования показывается движение "от точки к точке" в декартовой системе координат и связанные с ним команды ЧПУ.

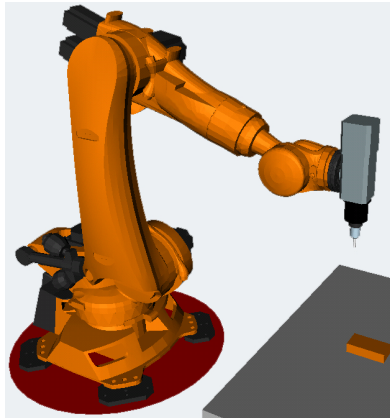


Рисунок 7-3 6-осевой робот со сгибающейся рукой и фрезерным шпинделем

```

N1 G90
N2 T="T8MILLD20" D1 M6
N3 TRAORI
; $P_UIFR[1]=CTrans(X,1500,Y,0,Z,400):CROT(X,0,Y,0,Z,-90)
N4 G54
N5 M3 S20000
N6 ORIWKS
N7 ORIVIRT1
N8 CYCLE832(0.01,_FINISH,1)
;HOME
N9 TRAFOOF
N10 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N11 TRAORI
N12 G54
N13 G0 PTP X1369.2426 Y956.7528 Z502.5517 A=135.5761 B=-33.2223 C=161.1435
STAT='B010' TU='B001011'
N14 G0 X1355.1242 Y1014.9394 Z424.9695 A=135.8491 B=-33.1439 C=160.9941
STAT='B010' TU='B001011'
N15 G1 CP X1354.8361 Y1016.1269 Z423.3862 A=136.0635 B=-33.0819 C=160.8770
F1000
N16 G1 X1336.4283 Y1016.1269 Z426.6311 A=136.0484 B=-32.2151 C=160.9643
F2000
N17 G1 X1317.9831 Y1016.1269 Z429.6730 A=136.0175 B=-31.3394 C=161.0655
;HOME
N18 TRAFOOF
N19 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N20 M30

```

7.10.5 Пример 2: Движение "от точки к точке" при базовой 5-осевой трансформации

Допущение: основой является прямоугольная кинематика СА.

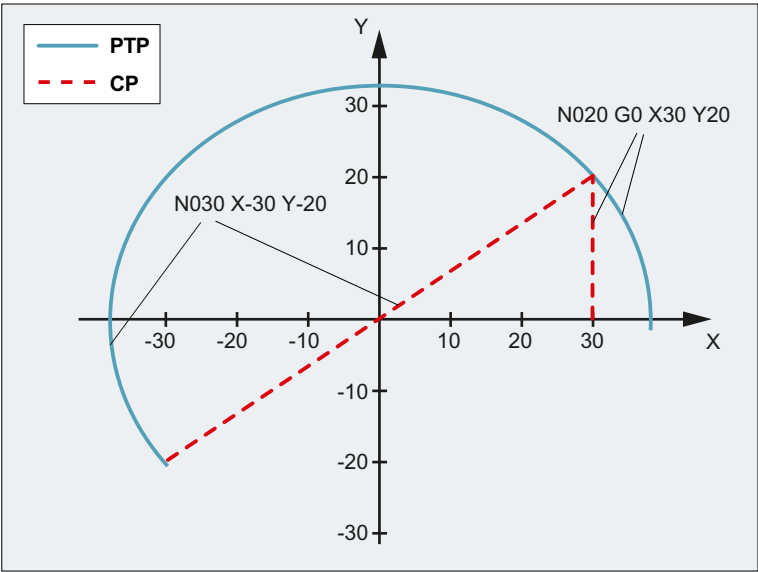
Программный код	Комментарий
TRAORI	; Трансформация кинематики СА вкл
PTP	; Включить движение "от точки к точке"
N10 A3=0 B3=0 C3=1	; позиция круговой оси C=0 A=0
N20 A3=1 B3=0 C3=1	; позиция круговой оси C=90 A=45
N30 A3=1 B3=0 C3=0	; позиция круговой оси C=90 A=90
N40 A3=1 B3=0 C3=1 STAT=1	; позиция круговой оси C=270 A=-45

Выбрать однозначное положение подвода к позиции круговых осей:

При этом в кадре N40 круговые оси через программирование **STAT=1** двигаются по более длинному пути от своей стартовой точки (C=90, A=90) к конечной точке (C=270, A=-45). При **STAT=0** круговые оси, напротив, двигаются к конечной точке (C=90, A=45) более коротким путем.

7.10.6 Пример 3: RTPG0 и

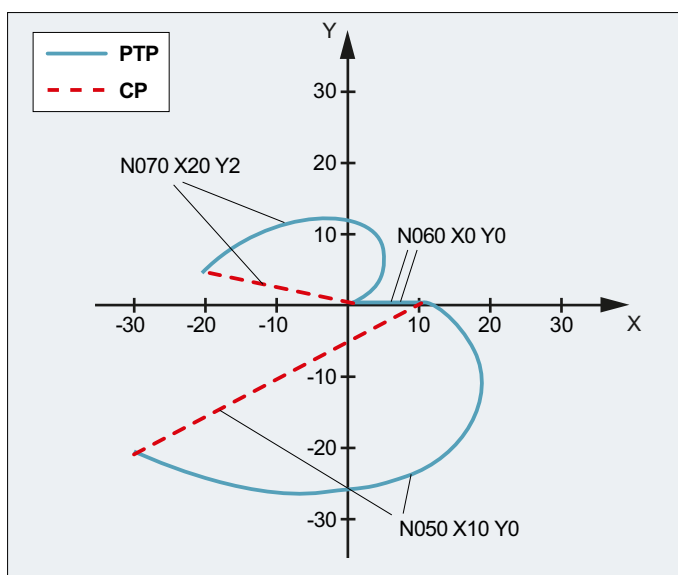
Обход полюса командами RTPG0 и



Программный код	Комментарий
N001 G0 X30 Z0 F10000 T1 D1 G90	; Исходная позиция, абсолютный размер
N002 SPOS=0	
N003 TRANSMIT	; трансформация TRANSMIT
N010 RTPG0	; к каждому G0-кадру автоматически RTP и затем снова CP.

Программный код	Комментарий
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Выход из полюса с PTPG0 и TRANSMIT



Программирование	Комментарий
N001 G0 X90 Z0 F10000 T1 D1 G90	; исходное положение
N002 SPOS=0	
N003 TRANSMIT	; трансформация TRANSMIT
N010 PTPG0	; к каждому G0-кадру автоматически PTP и затем снова CP.
N020 G0 X90 Y60	
N030 X-90 Y-60	
N040 X-30 Y-20	
N050 X10 Y0	
N060 X0 Y0	
N070 X-20 Y2	
N170 G1 X0 Y0	
N160 X10 Y0	
N150 X-30 Y-20	
M30	

7.11 Граничные условия при выборе трансформации

Функция

Выбор трансформаций возможен через программу обработки детали или MDA. При этом учитывать:

- Промежуточный кадр движения не вставляется (фаски/радиусы).
- Последовательность кадров сплайна должна быть завершена; если нет, то появляется сообщение.
- Точная коррекция инструмента должна быть отключена (FTOCOF); если нет, то появляется сообщение.
- Коррекция радиуса инструмента должна быть отключена (G40); если нет, то появляется сообщение.
- Активированная коррекция длин инструмента передается СЧПУ в трансформацию.
- Действующий перед трансформацией текущий фрейм отключается СЧПУ.
- Активное ограничение рабочего поля отключается СЧПУ для задействованных трансформацией осей (соответствует WALIMOF).
- Контроль защищенных областей отключается.
- Режим управления траекторией и перешлифовка прерываются.
- Все указанные в машинных данных оси должны быть синхронизированы с привязкой к кадру.
- Выполняется обратный переход перешедших осей, если нет, то появляется сообщение.
- Для зависимых осей выводится сообщение.

Смена инструмента

Смена инструмента допускается только при отмененной коррекции на радиус инструмента.

Смена коррекции длин инструмента и включение/выключение коррекции радиуса инструмента не могут программироваться в одном кадре.

Смена фрейма

Все операторы, относящиеся только к базовой кинематической системе, разрешены (FRAME, коррекция радиуса инструмента). Но смена фрейма при G91 (составной размер) – в отличие от не активной трансформации – не обрабатывается отдельно. Исполняемый инкремент обрабатывается в системе координат детали нового фрейма – независимо от того, какой фрейм действовал в предшествующем кадре.

Исключения

Затронутые трансформацией оси не могут использоваться:

- как ось Preset (аварийное сообщение),
- для подвода к фиксированной точке (ошибка),
- для реферирования (аварийное сообщение).

7.12 Отключение трансформации (TRAFOOF)

С помощью предопределенной процедуры TRAFOOF все активные трансформации и фреймы отключаются.

Примечание

Для отмены трансформации действуют те же граничные условия (Страница 402) что и для выбора.

Необходимые в дальнейшем фреймы должны быть активированы посредством повторного программирования.

Синтаксис

```
| ...  
| TRAFOOF
```

Значение

TRAFOOF:	Отключение всех активных трансформаций / фреймов
----------	--

Кинематические цепочки

8.1 Удаление компонентов (DELOBJ)

Функция `DELOBJ()` «удаляет» компоненты путем сброса назначенных системных переменных на значения по умолчанию:

- Элементы кинематических цепочек
- Защищенные области, элементы защищенных областей и соударяемые пары
- Данные трансформации

Синтаксис

```
[<RetVal>=] DELOBJ (<CompType> [ , , , <NoAlarm> ] )  
[<RetVal>=] DELOBJ (<CompType> , <Index1> [ , , <NoAlarm> ] )  
[<RetVal>=] DELOBJ (<CompType> [ , <Index1> ] [ , <Index2> ] [ , <NoAlarm> ] )
```

Значение

DELOBJ:	Удаление элементов кинематических цепочек, защищенных областей, элементов защищенных областей, соударяемых пар и данных трансформаций	
<CompType>:	Тип удаляемого компонента	
	Тип данных:	STRING
	Величина: "KIN_CHAIN_ELEM"	
	значение: Системные переменные всех кинематических элементов: \$NK_...	
	Величина: "KIN_CHAIN_SWITCH"	
	значение: Системная переменная \$NK_SWITCH[<i>]	
	Величина: "KIN_CHAIN_ALL"	
	Значение: Все кинематические элементы и коммутаторы. Равнозначно последовательному вызову DELOBJ с «KIN_CHAIN_ELEM» и «KIN_CHAIN_SWITCH»	
	Величина: «PROT_AREA»	
	значение: Системные переменные защищенных областей:	
	• \$NP_PROT_NAME • \$NP_CHAIN_NAME • \$NP_CHAIN_ELEM • \$NP_1ST_PROT	
	Величина: «PROT_AREA_ELEM»	
	значение: Системные переменные элементов защищенных областей станка и / или автоматических защищенных областей инструмента:	
	• \$NP_NAME • \$NP_NEXT • \$NP_NEXTP • \$NP_COLOR • \$NP_D_LEVEL • \$NP_USAGE • \$NP_TYPE • \$NP_FILENAME • \$NP_PARA • \$NP_OFF • \$NP_DIR • \$NP_ANG	
	Величина: «PROT_AREA_COLL_PAIRS»	
	значение: Системные переменные соударяемых пар:	
	• \$NP_COLL_PAIR • \$NP_SAFETY_DIST	
	Величина: «PROT_AREA_ALL »	
	значение: Все защищенные области, элементы защищенных областей и соударяемые пары (системные переменные \$NP_...) Равнозначно при последовательном вызове DELOBJ с «PROT_AREA», «PROT_AREA_ELEM» и «PROT_AREA_COLL_PAIRS»	
	Величина: «TRAFO_DATA»	
	значение: Системные переменные всех трансформаций \$NT_...	

<Index1>:	Индекс первого удаляемого компонента (опция)	
	Тип данных:	INT
	Значение по умолчанию:	-1
	Диапазон значений:	$-1 \leq x \leq$ (макс. число сконфигурированных компонентов -1)
	Значение	Значение
	0, 1, 2,	Индекс удаляемого компонента.
	-1	Все компоненты указанного типа удаляются. <Index2> не обрабатывается.
<Index2>:	Индекс последних удаляемых компонентов (опция) Если <Index2> не запрограммирован, то удаляются только системные переменные упомянутого в <Index1> компонента.	
	Тип данных:	INT
	Значение по умолчанию:	Удаляются только системные переменные упомянутого в <Index1> компонента.
	Диапазон значений:	$\text{<Index1>} < x \leq$ (макс. число сконфигурированных компонентов -1)
<NoAlarm>:	Подавление аварийных сообщений (опция)	
	Тип данных:	BOOL
	Значение по умолчанию:	false
	Значение	Объяснение
	false	В случае ошибки (<RetVal> < 0) выполнение программы останавливается, и сигнализируется аварийное сообщение..
	TRUE	В случае ошибки выполнение программы не останавливается и аварийное сообщение не отображается. Пример использования: Специализированная реакция согласно возвращаемому значению
<RetVal>:	Возвращаемое значение функции	
	Тип данных:	INT
	Диапазон значений:	0, -1, -2, ... -7
	Значение	Значение
	0	Нет ошибок.
	-1	Вызов функции без параметров. Как минимум, должен быть указан параметр <CompType>.
	-2	<CompType> обозначает неизвестный компонент
	-3	<Index1> меньше -1
	-4	<Index1> больше сконфигурированного числа компонентов
	-5	<Index1> имеет при удалении группы компонентов значение, отличное от -1
	-6	<Index2> меньше <Index1>
	-7	<Index2> больше сконфигурированного числа компонентов

8.2 Определение индекса по имени (NAMETOINT)

В поля системных переменных типа STRING внесены пользовательские имена. На основании идентификатора системных переменных и имени функция `NAMETOINT()` определяет относящееся к имени значение индекса, под которым оно записано в поле системной переменной.

Синтаксис

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

Значение

NAMETOINT:	Определение индекса системной переменной	
<SysVar>:	Имя поля системной переменной типа STRING	
	Тип данных:	STRING
	Диапазон значений:	Имена всех полей системной переменной ЧПУ типа STRING
<Name>:	Цепочка символов или имя, к которому должен быть найден индекс системной переменной.	
	Тип данных:	STRING
<NoAlarm>:	Подавление аварийных сообщений (опция)	
	Тип данных:	BOOL
	Значение по умолчанию:	FALSE
	Значение	Значение
	TRUE	В случае ошибки выполнение программы не останавливается и аварийное сообщение не отображается. Пример использования: Специализированная реакция согласно возвращаемому значению
	FALSE	В случае ошибки (<RetVal> < 0) выполнение программы останавливается, и сигнализируется аварийное сообщение.
<RetVal>:	Индекс системной переменной или сообщение об ошибке	
	Тип данных:	INT
	Диапазон значений:	$-1 \leq x \leq$ (макс. число сконфигурированных компонентов -1)
	Значение	Значение
	≥ 0	Искомое имя найдено под указанным индексом системной переменной.
	-1	Искомое имя не было найдено или возникла ошибка.

Пример

Программный код	Комментарий
DEF INT INDEX	
\$NP_PROT_NAME[27] = «кожух»	

Программный код	Комментарий
... INDEX = NAMETOINT («\$NP_PROT_NAME», «кожух»)	; INDEX == 27

Предотвращение столкновений с кинематическими цепочками

9

Примечание

Защищенные области

Названные в последующих главах защищенные области относятся к функции "Геометрическое моделирование станка"

Литература::

Описание функций - Специальные функции; глава "Геометрическое моделирование станка"

9.1 Проверка на соударяемую пару (COLLPAIR)

Функция `COLLPAIR(...)` определяет, образуют ли две защищенных области соударяемую пару.

Синтаксис

```
[<RetVal> =] COLLPAIR(<Name_1>,<Name_2>[,<NoAlarm>])
```

Объяснение

COLLPAIR:	Проверка на отношение к соударяемой паре		
<RetVal>:	Возвращаемое значение функции		
	Тип данных:	INT	
	Значение:	≥ 0	Обе защищенные области образуют соударяемую пару. Возвращаемое значение == индекс соударяемой пары m (см. \$NP_COLL_PAIR)
		-1	Либо было указано меньше двух строк, либо минимум одна из двух строк является нулевой.
		-2	Указанная в первом параметре защищенная область не найдена.
		-3	Указанная во втором параметре защищенная область не найдена.
		-4	Не найдена ни одна из двух указанных защищенных областей.
		-5	Обе указанные защищенные области были найдены, но не вместе в одной соударяемой паре.
<Name_1>:	Номер первой защищенной области		
	Тип данных:	STRING	
	Диапазон значений:	Спараметрированные имена защищенных областей	
<Name_2>:	Имя второй защищенной области		
	Тип данных:	STRING	
	Диапазон значений:	Спараметрированные имена защищенных областей	
<NoAlarm>:	Подавление аварийных сообщений (опционально)		
	Тип данных:	BOOL	
	Значение:	FALSE (по умолчанию)	В случае ошибки (<RetVal> < 0) выполнение программы останавливается и сигнализируется аварийное сообщение.
		TRUE	В случае ошибки выполнение программы не останавливается и аварийное сообщение не отображается. Пример использования: Специализированная реакция согласно возвращаемому значению

9.2 Запрос повторного вычисления модели станка для предотвращения столкновений (PROTA)

Если системная переменная кинематической цепи \$NK_..., геометрического моделирования станка или предотвращения столкновений \$NP_... описываются в программе обработки детали, то затем необходимо вызвать процедуру PROTA, чтобы принять изменение модели станка для предотвращения столкновений внутри ЧПУ.

Синтаксис

PROTA [(<Par>)]

Объяснение

PROTA:	Запрос повторного вычисления модели станка для предотвращения столкновений		
	- Вызывает остановку предварительной обработки. - Должен стоять автономно в кадре.		
<Par>:	Параметры (опционально)		
	Тип данных:	STRING	
	Значение:	---	Без параметров. Выполняется повторный расчет модели станка. Состояния защищенных областей сохраняются.
		"R"	Выполняется повторный расчет модели станка. Защищенные области приводятся в состояние инициализации согласно \$NP_INIT_STAT.

Граничные условия

Моделирование

Процедуру PROTA нельзя использовать в программах обработки детали в сочетании с моделированием (simNC).

Пример: Предотвращение вызова PROTA, , пока моделирование активно.

Программный код	Комментарий
...	
IF \$P_SIM == FALSE	; IF моделирование не активно
PROTA	; THEN заново рассчитать модель столкновений
ENDIF	; ENDIF
...	

См. также

Установка состояния защищенной области (PROTS) (Страница 414)

9.3 Установка состояния защищенной области (PROTS)

Процедура PROTS (. . .) устанавливает указанное значение состояния защищенных областей.

Синтаксис

```
PROTS (<State>[, <Name_1>, ..., <Name_n>])
```

Объяснение

PROTS:	Установка состояния защищенных областей		
	● Должна стоять автономно в кадре.		
<State>:	Состояние, на которое должны быть установлены указанные защищенные области		
	Тип данных:	CHAR	
	Значение:	"A" или "a"	Состояние: активна
		"I" или "i"	Состояние: не активна
		"P"или "p"	Состояние: предварительно активировано или управляется через PLC ¹⁾
"R"или "r"		Состояние: Внутреннее для ЧПУ значение состояния инициализации ²⁾	
<Name_1> ... <Name_n>:	Имя одной или нескольких защищенных областей, которые должны быть установлены на указанное состояние (опционально) Если имя не указано, то указанное состояние устанавливается для всех определенных защищенных областей.		
	Тип данных:	STRING	
	Диапазон значений:	Спараметрированные имена защищенных областей	
	Примечание: Максимальное число защищенных областей, которые могут быть указаны как параметры, зависит только от максимально возможного числа символов на строку программы.		

¹⁾ Активация / деактивация осуществляется через: DB10.DBX234.0 - DBX241.7

²⁾ Для состояния устанавливается внутреннее для ЧПУ значение состояния инициализации, т. е. значение, которое системная переменная \$NP_INIT_STAT имела на момент последнего вызова PROTA(...) (Страница 413).

9.4 Определение интервала между двумя защищенными областями (PROTD)

Функция `PROTD ( . . . )` вычисляет отступ двух защищенных областей.

Свойства функции:

- Расчет интервала выполняется независимо от состояния защищенных областей (активированы, деактивированы, предактивированы).
- Для вычисления отступа двух защищенных областей используются только элементы защищенных областей, обозначенные `$NP_USAGE = "C"` или `"A"`. Элементы защищенной области, обозначенные посредством `$NP_USAGE = "V"`, не учитываются.
- Защищенные области, в которых элементы защищенной области обозначены посредством `$NP_USAGE = "V"`, не могут быть использованы для вычисления отступа.
- Расчет интервала выполняется с действительными на конце предшествующего кадра позициями.
- Наложения, учитываемые при главном ходе (например, DRF-смещение или внешние смещения нулевой точки) включаются в расчет отступа с действующими на **момент интерполяции** функции значениями.

Примечание

Синхронизация

При использовании функции `PROTD ( . . . )` ответственность за возможно необходимую синхронизацию главного хода и предварительной обработки с помощью остановки предварительной обработки `STOPRE` лежит только на пользователе.

Столкновение

Если между указанными защищенными областями имеет место столкновение, то функция возвращает отступ 0,0. Столкновение имеет место, если обе защищенные области соприкасаются или имеют наложение.

Безопасное расстояние для проверки столкновений (`MD10622 $MN_COLLISION_SAFETY_DIST`) при вычислении отступа не учитывается.

Синтаксис

```
[<RetVal> =] PROTD ([<Name_1>], [<Name_2>], VAR <Vector>[, <System>])
```

Объяснение

PROTD:	Рассчитывает отступ между двумя указанными защищенными областями
	• Должна стоять автономно в кадре.

9.4 Определение интервала между двумя защищенными областями (PROTD)

<RetVal>:	Возвращаемое значение функции: Абсолютное значение интервала между двумя защищенными областями или 0,0 при столкновении (см. выше: раздел Столкновение)		
	Тип данных:	REAL	
	Диапазон значений:	0,0 ≤ x ≤ +макс. REAL-значение	
<Name_1>, <Name_2>:	Имена обеих защищенных областей, отступ между которыми необходимо вычислить (опционально)		
	Тип данных:	STRING	
	Диапазон значений:	Спараметрированные имена защищенных областей	
	Значение по умолчанию:	"" (пустая строка) Если защищенные области не указаны, то функция вычисляет наименьшее текущее расстояние из всех в содержащихся в модели столкновений активированных и предактивированных защищенных областей.	
<Vector>:	Возвращаемое значение: 3-мерный вектор расстояния от защищенной области <Name_2> к защищенной области <Name_1> с: • <Vector>[0]: X-координата в мировой системе координат • <Vector>[1]: Y-координата в мировой системе координат • <Vector>[2]: Z-координата в мировой системе координат При столкновении: <Vector> == нулевой вектор		
	Тип данных:	VAR REAL [3]	
	Диапазон значений:	<Vector> [n]: 0,0 ≤ x ≤ ±макс. REAL-значение	
	<System>:	Система единиц (дюймовая / метрическая) для интервала и вектора расстояния (опция)	
Тип данных:		BOOL	
Значение:		FALSE (по умолчанию)	Система единиц согласно текущей активной G-команде из G-группы 13 (G70, G71, G700, G710).
		TRUE	Система единиц согласно установленной основной системе: MD52806 \$MN_ISO_SCALING_SYSTEM

Трансформации с кинематическими цепями

10.1 Включение трансформации (TRAFOON)

Трансформацию, определенную посредством кинематических цепей, активирует предварительно определенная процедура TRAFOON. В кадре должен находиться только вызов.

Примечание

В качестве альтернативы трансформацию, определенную посредством кинематических цепей, можно активировать и традиционными языковыми командами, например, TRAORI или TRANSMIT. Для этого в системной переменной \$NT_TRAFO_INDEX должно быть внесено соответствующее значение, не равное нулю.

Дополнительная информация по \$NT_TRAFO_INDEX представлена в "Справочник таблиц — Системные переменные".

Синтаксис

TRAFOON (<Trafoname>, <Diameter>, <k>)

Значение

TRAFOON:	Процедура активации трансформации, определенной посредством кинематических цепей	
<Trafoname>:	имя блока данных трансформации	
	Тип данных:	STRING
	Диапазон значений:	все определенные посредством \$NT_NAME имена блоков данных трансформаций
	Указание: Имя блока данных трансформации должно быть однозначным. Оно должно встречаться в \$NT_NAME только один раз.	
<Diameter>:	базовый или рабочий диаметр (только TRACYL)	
	Тип данных:	REAL
	Значение должно быть > 1.	
<k>:	определяет использование коррекции стенки паза (только TRACYL).	
	Тип данных:	BOOL
	Значение:	FALSE без коррекции стенки паза
		TRUE с коррекцией стенки паза
	Соответствует типу трансформации 514 TRACYL (коррекция стенки паза программируемая). Если <k> не задано, действует параметризованная настройка бита 10 в \$NT_CNTRL[<n>].	

Пример

Программный код	Комментарий
TRAFOON["Trans_1"]	Активирует трансформацию под именем Trans_1.

10.2 Модифицирование трансформации ориентации после замеров станка (CORRTRAFO)

В станках с трансформациями ориентации, которые были определены посредством кинематических цепей, пользователь может использовать предварительно определенную функцию CORRTRAFO для модифицирования векторов смещения или векторов направления осей ориентации в кинематической модели станка после замеров станка.

Синтаксис

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode> [,  
<No_Alarm>])
```

Значение

CORRTrafo:	Вызов функции
------------	---------------

10.2 Модифицирование трансформации ориентации после замеров станка (CORRTRAFO)

<Corr_Status>:	Возвращаемое значение функции	
	Тип данных:	INT
	Значения:	0 Функция была выполнена без ошибок.
		1 Ни одна из трансформаций не активна.
		2 Активная на данный момент трансформация не является трансформацией ориентации.
		3 Активная трансформация ориентации не была определена посредством кинематических цепей.
		10 Параметр вызова <Corr_Index> отрицательный.
		11 Параметр вызова <Corr_Mode> отрицательный.
		12 Недействительная ссылка на сегмент подцепочки (1-я позиция <Corr_Index>). Значение не должно быть больше значения количества осей ориентации в подцепочке.
		13 Недействительная ссылка на ось ориентации подцепочки (1-я позиция <Corr_Index>). Значение должно быть меньше значения количества осей ориентации в подцепочке.
		14 Недействительная ссылка на подцепочку (10-я позиция <Corr_Index>). Допустимы только значения 0 и 1 (ссылка на цепь детали или цепь инструмента). Этот номер ошибки отображается также, если подцепочка, на которую ссылается <Corr_Index>, не существует.
		15 На сегменте, на который выполняется ссылка параметром <Corr_Index>, элемент коррекции не определен (\$NT_CORR_ELEM_P или \$NT_CORR_ELEM_T).
		20 Недействительный режим коррекции (1-я позиция <Corr_Mode>). Допустимы только значения 0 и 1.
		21 Недействительный режим коррекции (10-я и/или 100-я позиция <Corr_Mode>). При записи осевого направления только 1-я позиция должна быть не равна нулю.
		30 100-я позиция <Corr_Mode> недействительна. Допустимы только значения 0 и 1.
		31 1000-я позиция <Corr_Mode> недействительна. Допустимы только значения 0 и 1.
		40 Вектор направления, который необходимо принять как осевое направление, является нулевым вектором. Эта ошибка может возникнуть, только если 1000-я позиция <Corr_Mode> равна 0. Если 1000-я позиция этого параметра равна 1 (контроль максимальной коррекции деактивирован), может быть записан только нулевой вектор.
		41 При коррекции вектора смещения отклонение по сравнению с текущим значением, по меньшей мере, в одной координате больше максимального значения, заданного в установочных данных SD41610 \$SN_CORR_TRAFO_LIN_MAX. Параметр <Corr_Vect> перезаписывается через вектор ошибки. То же действительно, и если обработку прервало аварийное сообщение (см. параметр <No_Alarm>). Вектор ошибки содержит в компонентах, значение коррекции которых превысило допустимый предел, разницу (с

			учёт знака) установленного значения коррекции и пред-ела. Содержание компонентов, не превысивших свой предел, равно нулю.
		42	При коррекции вектора направления угловая погрешность по сравнению с текущим направлением больше макси-мального значения, заданного в установочных данных SD41611 \$SN_CORR_TRAFO_DIR_MAX.
		43	Попытка записи системной переменной была отклонена из-за отсутствия прав записи.
<Corr_Vect>:	Вектор коррекции Содержание вектора коррекции определяется последующими параметра-ми <Corr_Index> и <Corr_Mode>. Если <Corr_Status> = 41, содержание вектора перезаписывается (см. выше).		
	Тип дан-ных:	REAL	
<Corr_Index>:	Сегмент, элемент коррекции которого необходимо модифицировать / Индекс оси ориентации, вектор направления которой необходимо модифи-цировать		
	Тип дан-ных:	INT	
	Параметр <Corr_Index> записан в десятичном коде (с 1-й позиции по 10-ю позицию).		
	1-я позиция:	Содержит индекс сегмента или оси ориентации в подцепочке.	
	10-я позиция:	Ссылается на подцепочку.	
		0x	Цепь детали
	1x	Цепь инструмента	

10.2 Модифицирование трансформации ориентации после замеров станка (CORRTRAFO)

<Corr_Mode>:	Режим коррекции	
	Тип данных:	INT
	Параметр <Corr_Mode> записан в десятичном коде (с 1-й позиции по 1000-ю позицию).	
	1-я позиция:	Определяет, который элемент необходимо корректировать.
		xxx0 Коррекция линейного вектора смещения
		xxx1 Коррекция вектора направления оси ориентации
	10-я позиция:	Определяет, каким образом необходимо модифицировать элемент коррекции, на который ссылается содержание <Corr_Index>.
		xx0x Вектор коррекции записывается непосредственно в элемент коррекции. Этот вариант можно использовать для непосредственного описания элемента коррекции, причем не требуется знать индекс <n> соответствующих системных данных (\$NK_OFF_DIR[<n>, ...]).
		xx1x Как 0, но с тем отличием, что переданное значение коррекции интерпретируется в мировой системе координат. Разница между вариантами 0 и 1 всегда возникает в случае, если кинематическая цепь содержит в исходном положении (позиции всех осей ориентации равны 0) дополнительные вращения.
		xx2x Как 1, но с тем отличием, что значение коррекции относится ко всему сегменту, т. е. в элемент коррекции вносится такое значение, что весь сегмент достигает длины, определенной посредством значения коррекции.
		Указание: Для описания вектора направления оси ориентации значения 1 и 2 не разрешены.
	100-я позиция:	Определяет, как следует интерпретировать содержание параметра <Corr_Vect>.
		x0xx Переданный вектор коррекции <Corr_Vect> содержит всю новую длину элемента коррекции или сегмента, на который ссылается <Corr_Index> в сочетании с 10-й позицией <Corr_Mode> (абсолютная коррекция).
		x1xx Переданный вектор коррекции <Corr_Vect> содержит только разницу по отношению к длине элемента коррекции или сегмента, на который ссылается <Corr_Index> в сочетании с 10-й позицией <Corr_Mode> (инкрементальная коррекция).
		Указание: При коррекции вектора направления оси ориентации содержание 100-й позиции должно быть 0.
	1000-я позиция:	Определяет, необходимо ли ограничивать коррекцию посредством следующего максимального значения:
		- SD41610 \$SN_CORR_TRAFO_LIN_MAX - или - SD41611 \$SN_CORR_TRAFO_DIR_MAX
	0xxx	Контроль максимальной коррекции активен.

		1xxx	Контроль максимальной коррекции не активен.
<No_Alarm>:	Поведение в случае аварийного сообщения (возвращаемое значение > 0) (опционально)		
	Тип данных:	BOOL	
	Значение:	FALSE (по умолчанию)	В случае ошибки выполнение программы приостанавливается, и отображается аварийное сообщение 14103.
		TRUE	В случае ошибки выполнение программы не останавливается и аварийное сообщение не отображается. Пример использования: Специализированная реакция согласно возвращаемому значению

Примечание

Если при вызове функции появляется ошибка, то либо выводится аварийное сообщение, либо возвращается номер ошибки (см. Параметр <No_Alarm>), так что пользователь может самостоятельно надлежащим образом отреагировать на неполадку. Причина ошибки подробнее обозначена в параметре аварийного сообщения. Возвращаемый вместо аварийного сообщения номер ошибки идентичен параметру аварийного сообщения.

Дополнительная информация по CORRTRAFO

Кинематическая структура станка с трансформацией ориентации описывается на основе одной или двух кинематических цепей (подцепей), исходящих из нулевой точки мировой системы координат. Одна из этих цепей — **цепь инструмента** — заканчивается в исходной точке инструмента, другая — **цепь детали** — в нулевой точке базовой кинематической системы (BKS).

Функция CORRTRAFO измеряет длины плеча рычага и осевые направления в станках с трансформацией ориентации и записывает эти значения в специальные элементы коррекции. Кинематическая цепь описывается, кроме прочего, посредством элементов типа OFFSET, определяемых посредством \$NK_TYPE.

CORRTRAFO работает с сегментами.

Обе подцепочки могут подразделяться не более чем на четыре сегмента:

- Сегмент 1 начинается в стартовой точке цепочки и заканчивается на первой оси ориентации.
- Сегмент 2 – сегмент между осью ориентации 1 и осью ориентации 2.
- Сегмент 3 – сегмент между осью ориентации 2 и осью ориентации 3.
- Сегмент 4 – сегмент между осью ориентации 3 и концом цепи инструмента или детали.

Сегменты могут содержать соответственно постоянные элементы цепочки типов OFFSET или ROT_CONST.

На следующем рисунке показана трансформация ориентации с 2 осями ориентации.

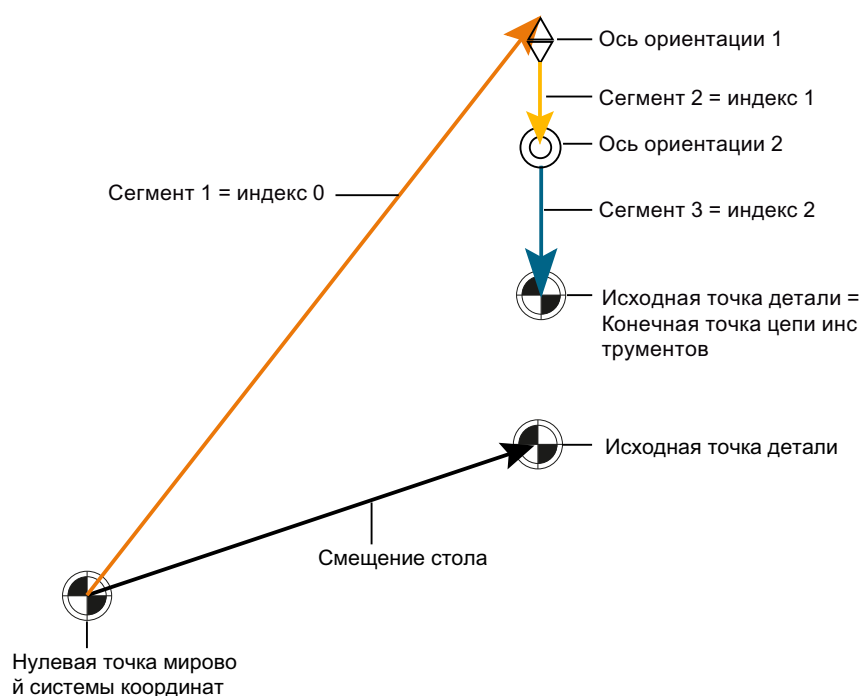


Рисунок 10-1 Пример CORRTRAFO

Сегменты имеют однозначное определение: При следовании по кинематической подцепочке от ее стартовой точки до конечной точки получается, что у первого сегмента — индекс 0, у следующего сегмента — индекс 1 и т. п. В таком случае индекс последнего сегмента всегда равен количеству осей ориентации.

Элементы коррекции

На каждый постоянный элемент кинематической цепи (элемент цепи типа `$NK_TYPE[<n>] = "OFFSET"`) в каждом из этих сегментов можно ссылаться посредством системной переменной `$NT_CORR_ELEM_T[<n>, 0 ... 3]` или `$NT_CORR_ELEM_P[<n>, 0 ... 3]`. При помощи функции CORRTRAFO в обозначенные таким образом элементы записываются значения коррекции, установленные при замерах станка.

Пример с индексом трансформации = 1:

- `$NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"`; смещение оси C (ось ориентации 1) в сегменте 1 определено как элемент коррекции.
- `$NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"`; смещение оси B (ось ориентации 2) в сегменте 2 определено как элемент коррекции.
- `$NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"`; смещение оси B к исходной точке инструмента в сегменте 3 определено как элемент коррекции.

Последовательность ссылок в `$NT_CORR_ELEM_T/P[<n>, 0 ... 3]` должна соответствовать описанным выше сегментам, т. е., в `$NT_CORR_ELEM_T/P[<n>, 0]` может находиться только элемент цепи, расположенный перед первой осью ориентации, и т. п.

Функция CORRTRAFO записывает в определенные таким образом элементы коррекции значения, которые были определены посредством замеров станка. Модифицирование

значений коррекции определяется в CORRTRAFO посредством параметра `<Corr_Mode>`.

Закрытие цепи

Если в системной переменной `$NT_CNTRL[<n>]` установлены бит 7 или бит 8, то в конце цепи детали (бит 7) или перед стартовой точкой цепи инструмента (бит 8) внутри автоматически добавляются дополнительные постоянные элементы цепи, создающие связь от конечной точки цепи к нулевой точке станка ("Закрытие цепи").

Извне эти автоматически добавленные элементы можно только читать, запись в них невозможна (см. системные переменные `$AC_TRAFO_CORR_ELEM_P/T`).

Точка закрытия цепи инструментов

Если системная переменная `$NT_CLOSE_CHAIN_T` не пустая, цепь инструмента закрывается не в конце цепи, а в конечной точке обозначенного элемента цепи. И тогда другие элементы цепи, находящиеся позади этой точки, приводят при активации трансформации к соответствующему смещению нулевой точки.

Индекс оси ориентации

Кроме постоянного смещения между осями ориентации, функция CORRTRAFO может описывать и векторы направления осей ориентации. Под индексом оси ориентации в данном случае понимают индекс, результирующий, если пройти кинематическую подцепочку с начала до её конца, при этом счёт начинается с нуля. Поэтому индекс оси ориентации всегда равен индексу расположенного перед ней сегмента.

Индекс оси ориентации можно определить и посредством системной переменной `$AC_TRAFO_ORIAX_LOC`.

Максимально допустимое изменение элемента цепи

Максимально допустимое изменение элемента цепи можно ограничить обоими установочными данными `SD41610 $SN_CORR_TRAFO_LIN_MAX` для векторов смещения и `SD41611 $SN_CORR_TRAFO_DIR_MAX` для векторов направления осей ориентации. `SD41610 $SN_CORR_TRAFO_LIN_MAX` задает, на какую максимальную величину разрешено изменять каждый отдельный компонент вектора по сравнению с его исходным значением. `SD41611 $SN_CORR_TRAFO_DIR_MAX` задает, на какой максимальный угол разрешено изменять направление вектора оси по сравнению с его исходным значением. При этом исходное значение — это значение, действующее в трансформации, активной при вызове CORRTRAFO. Это означает, что, возможно, измененное при активации трансформации содержание данных кинематики в системе УД не влияет на действие функции CORRTRAFO.

Коррекции инструмента

11.1 Память коррекций

Структура памяти коррекций

Любое поле данных может быть вызвано с помощью номера T и D (кроме "плоских D-№") и содержит, наряду с геометрическими данными для инструмента, и другие элементы, например, тип инструмента.

Плоская структура номеров D

"Плоская структура номеров D" используется, если управление инструментом осуществляется вне ЧПУ. В этом случае номера D создаются с соответствующими кадрами коррекции инструмента без согласования с инструментами.

В программе обработки детали может продолжаться программироваться T. Но этот T не связан с запрограммированным номером D.

Данные резцов пользователя

Через машинные данные могут быть сконфигурированы данные резцов пользователя. Следовать указаниям изготовителя станка.

Параметры инструмента

Примечание

Отдельные значения в памяти коррекций

Отдельные значения памяти коррекций P1 до P25 через системные переменные могут считываться и записываться из программы. Все прочие параметры зарезервированы.

Параметры инструмента \$TC_DP6 до \$TC_DP8, \$TC_DP10 и \$TC_DP11, а также \$TC_DP15 до \$TC_DP17, \$TC_DP19 и \$TC_DP20, в зависимости от типа инструмента, имеют иное значение.

Номер параметра инструмента (DP)	Значение системных переменных	Примечание
\$TC_DP1	Тип инструмента	Обзор см. Список
\$TC_DP2	Положение резцов	Только для токарных инструментов
Геометрия	Коррекция длин	
\$TC_DP3	Длина 1	Вычисление по
\$TC_DP4	Длина 2	типу и плоскости
\$TC_DP5	Длина 3	
Геометрия	Радиус	

Номер параметра инструмента (DP)	Значение системных переменных	Примечание
\$TC_DP6 ¹⁾ \$TC_DP6 ²⁾	Радиус 1 / длина 1 Диаметр d	Фрезерный/токарный/шлифовальный инструмент Наградка
\$TC_DP7 ¹⁾ \$TC_DP7 ²⁾	Длина 2 / угловой радиус конической фрезы Ширина паза b Угловой радиус	Фрезерные инструменты Наградка
\$TC_DP8 ¹⁾ \$TC_DP8 ²⁾	Радиус закругления 1 для фрезерных инструментов Выступ k	Фрезерные инструменты Наградка
\$TC_DP9 ^{1) 3)}	Радиус закругления 2	зарезервировано
\$TC_DP10 ¹⁾	Угол 1 торцовая сторона инструмента	Конические фрезерные инструменты
\$TC_DP11 ¹⁾	Угол 2 продольная ось инструмента	Конические фрезерные инструменты
Износ	Коррекция длин и радиуса	
\$TC_DP12	Длина 1	
\$TC_DP13	Длина 2	
\$TC_DP14	Длина 3	
\$TC_DP15 ¹⁾ \$TC_DP15 ²⁾	Радиус 1 / длина 1 Диаметр d	Фрезерный/токарный/шлифовальный инструмент Наградка
\$TC_DP16 ¹⁾ \$TC_DP16 ³⁾	Длина 2 / угловой радиус конической фрезы Ширина паза b Угловой радиус	Фрезерные инструменты Наградка
\$TC_DP17 ¹⁾ \$TC_DP17 ²⁾	Радиус закругления 1 для фрезерных инструментов Выступ k	Фрезерование / 3D торцовое фрезерование Наградка
\$TC_DP18 ^{1) 3)}	Радиус закругления 2	зарезервировано
\$TC_DP19 ¹⁾	Угол 1 торцовая сторона инструмента	Конические фрезерные инструменты
\$TC_DP20 ¹⁾	Угол 2 продольная ось инструмента	Конические фрезерные инструменты
Базовый размер/адаптер	Коррекции длин	
\$TC_DP21	Длина 1	
\$TC_DP22	Длина 2	
\$TC_DP23	Длина 3	
Технология		
\$TC_DP24	Задний угол	Только для токарных инструментов
\$TC_DP25		зарезервировано

¹⁾ Действительно и для фрезерных инструментов для 3D-торцового фрезерования

²⁾ Для типа инструмента "наградка"

³⁾ зарезервировано (не используется SINUMERIK 840D sl)

Примечания

Для геометрических величин (например, длина 1 или радиус) существует несколько компонентов записи. Они аддитивно вычисляются в результирующую величину (например, общая длина 1, общий радиус), которая после активируется.

Не нужным коррекциям присваивается значение ноль.

Параметры инструмента \$TC-DP1 до \$TC-DP23 с контурными инструментами**Примечание**

Параметры инструментов, не включенные в таблицу, например, \$TC_DP7, не обрабатываются, т. е. их содержание не имеет значения.

Номер параметра инструмента (DP)	Объяснение	Резцы Dn		Примечание
\$TC_DP1	Тип инструмента			400 до 599
\$TC_DP2	Положение резцов			
Геометрия	Коррекция длин			
\$TC_DP3	Длина 1			
\$TC_DP4	Длина 2			
\$TC_DP5	Длина 3			
Геометрия	Радиус			
\$TC_DP6	Радиус			
Геометрия	Предельный угол			
\$TC_DP10	Мин. предельный угол			
\$TC_DP11	Макс. предельный угол			
Износ	Коррекция длин и радиуса			
\$TC_DP12	Износ длина 1			
\$TC_DP13	Износ длина 2			
\$TC_DP14	Износ длина 3			
\$TC_DP15	Износ радиус			
Износ	Предельный угол			
\$TC_DP19	Износ мин. предельный угол			
\$TC_DP20	Износ макс. предельный угол			
Базовый размер/адаптер	Коррекции длин			
\$TC_DP21	Длина 1			
\$TC_DP22	Длина 2			
\$TC_DP23	Длина 3			

Основное значение и значение износа

Результирующие величины получаются как сумма основного значения и значения износа соответственно (например, \$TC_DP6 + \$TC_DP15 для радиуса). К длине инструмента первого резца кроме этого прибавляется и базовый размер (\$TC_DP21 –

\$TC_DP23). Дополнительно на эту длину инструмента действуют все другие величины, которые и у обычного инструмента могут влиять на эффективную длину инструмента (адаптер, ориентируемый инструментальный суппорт, установочные данные).

Предельный угол 1 и 2

Предельные углы 1 или 2 относятся к вектору от центра резца к исходной точке резца соответственно и подсчитываются против часовой стрелки.

11.2 Аддитивные коррекции

11.2.1 Выбор аддитивных коррекций (DL)

Аддитивные коррекции могут рассматриваться как программируемые при обработке коррекции процесса. Они относятся к геометрическим данным резца, являясь тем самым составной частью данных резцов инструмента.

Обращение к данным аддитивной коррекции осуществляется через номер DL (DL: Location dependent; коррекции относительно соответствующего места использования) и они вводятся через интерфейс.

Использование

Через аддитивные коррекции возможно исправление обусловленных местом использования погрешностей размера.

Синтаксис

DL=<номер>

Значение

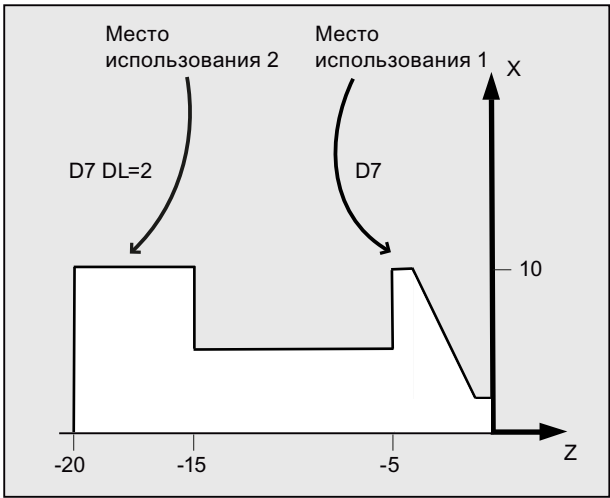
DL:	Команда для активации аддитивной коррекции
<номер>:	Через параметр <номер> указывается активируемый аддитивный блок данных коррекции инструмента

Примечание

Определение числа и активация аддитивных коррекций осуществляется через машинные данные (→ Следовать указаниям изготовителя станка!).

Пример

Один и тот же резец используется для 2 гнезд подшипника:



Программный код	Комментарий
N110 T7 D7	; Револьвер позиционируется на место 7. D7 и DL=1 активируются и выводятся в следующем кадре.
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; Аддитивно к D7 активируется DL=2 и выводится в следующем кадре.
N150 G1 Z-21	
N160 G0 X200 Z200	; Подвод к точке смены инструмента.
...	

11.2.2
 Определение значений износа и установочных значений (\$TC_SCPxy[t,d], \$TC_ECPxy[t,d])

Значения износа и установочные значения могут считываться и записываться через системные переменные. При этом логика ориентируется на логику соответствующих системных переменных для инструментов и резцов.

Системные переменные

\$TC_SCPxy[<t>,<d>]:	Значения износа, согласованные через ху с соответствующим геометрическим параметром, при этом х соответствует номеру значения износа, а у создает связь с геометрическим параметром.
\$TC_ECPxy[<t>,<d>]:	Установочные значения, согласованные через ху с соответствующим геометрическим параметром, при этом х соответствует номеру установочного значения, а у создает связь с геометрическим параметром.
<t>: Т-номер инструмента	
<d>: номер D резца инструмента	

Примечание

Определенные значения износа и установочные значения прибавляются к геометрическим параметрам и прочим параметрам коррекции (номер D).

Пример

Значение износа длины 1 устанавливается для резца <d> инструмента <t> на значение 1.0.

Параметры: \$TC_DP3 (длина 1, для токарных инструментов)

Значения износа: от \$TC_SCP13 до \$TC_SCP63

Установочные значения: от \$TC_ECP13 до \$TC_ECP63

\$TC_SCP43 [<t>,<d>] = 1.0

11.2.3 Удаление аддитивных коррекций (DELDL)

С помощью команды DELDL аддитивные коррекции для резца инструмента удаляются (освобождение памяти). При этом удаляются как определенные значения износа, так и установочные значения.

Синтаксис

```
DELDL [<t>,<d>]
DELDL [<t>]
DELDL
<состояние>=DELDL [<t>,<d>]
```

Значение

DELDL:	Команда для удаления аддитивных коррекций	
<t>:	номер T инструмента	
<d>:	номер D резца инструмента	
DELDL [<t>,<d>]:	Удаляются все аддитивные коррекции резца <d> инструмента <t>.	
DELDL [<t>]:	Удаляются все аддитивные коррекции всех резцов инструмента <t>.	
DELDL:	Удаляются все аддитивные коррекции всех резцов всех инструментов блока ТО (для канала, в котором программируется команда).	
<состояние>:	Состояние удаления	
	Величина:	Значение:
	0	Удаление было выполнено успешно.
	-	Удаление не было выполнено (если параметрирование обозначает точно один резец), или удаление было выполнено не полностью (если параметрирование обозначает несколько резцов).

Примечание

Значения износа и установочные значения активных инструментов не могут быть удалены (поведение аналогично удалению D или данных инструмента).

11.3 Коррекция инструмента - специальная обработка

С помощью установочных данных SD42900 bis SD42960 можно управлять нормированием знаков для длины инструмента и износа.

Это же относится к поведению компонентов износа при отражении гео-осей или при смене плоскости обработки, а также к температурной компенсации в направлении инструмента.

Значения износа

Если ниже имеются ссылки на значения износа, то под ними понимается сумма самих значений износа (\$TC_DP12 до \$TC_DP20) и суммарных коррекций со значениями износа (\$SCPX3 до \$SCPX11) и установочными значениями (\$ECPX3 до \$ECPX11).

Подробности по суммарным коррекциям см.:

Литература:

Описание функций "Управление инструментом"

Установочные данные

SD42900 \$SC_MIRROR_TOOL_LENGTH	Отражение компонентов длин инструмента и компонентов базового размера.
SD42910 \$SC_MIRROR_TOOL_WEAR	Отражение значений износа компонентов длин инструмента.
SD42920 \$SC_WEAR_SIGN_CUTPOS	Нормирование знака компонентов износа в зависимости от положения резца.
SD42930 \$SC_WEAR_SIGN	Инвертирует знаки размеров износа.
SD42935 \$SC_WEAR_TRANSFORM	Преобразование значений износа.
SD42940 \$SC_TOOL_LENGTH_CONST	Согласование компонентов длин инструмента с гео-осями.
SD42950 \$SC_TOOL_LENGTH_TYPE	Согласование компонентов длин инструмента независимо от типа инструмента.
SD42960 \$SC_TOOL_TEMP_COMP	Значение температурной компенсации в направлении инструмента. Действует и при наличии ориентации инструмента.

Литература

Описание функций - Основные функции; Коррекция на инструмент (W1)

Дополнительная информация

Активация измененных установочных данных

Новое нормирование компонентов инструмента при изменении описываемых установочных данных активируется только при следующем выборе резца инструмента. Если инструмент уже активен и необходимо активировать измененное нормирование данных этого инструмента, то требуется заново выбрать этот инструмент.

Это же относится и к случаю, когда изменяется результирующая длина инструмента, т.к. было изменено состояние отражения оси. Для активации измененных компонентов длин инструмента необходимо заново выбрать инструмент после команды отражения.

Ориентируемые инструментальные суппорты и новые установочные данные

Установочные данные SD42900 до SD42940 не действуют на компоненты возможно активного ориентируемого инструментального суппорта. Однако инструмент со всей его полученной длиной (длина инструмента + износ + базовый размер) всегда включается в вычисление с ориентируемым инструментальным суппортом. При расчете результирующей общей длины учитываются все изменения, вызванные установочными данными; т. е. векторы ориентируемого инструментального суппорта не зависят от плоскости обработки.

Примечание

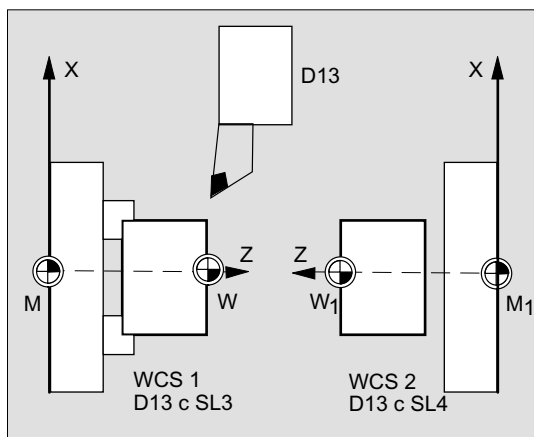
Часто при использовании ориентируемого инструментального суппорта имеет смысл определить все инструменты для не отраженной основной системы, включая те, которые используются только при зеркальной обработке. Тогда при обработке с отраженными осями инструментальный суппорт поворачивается таким образом, что фактическое положение инструмента описывается правильно. В этом случае все компоненты длин инструмента автоматически действуют в правильном направлении, поэтому управление нормированием отдельных компонентов через установочные данные в зависимости от состояния отражения отдельных осей становится ненужным.

Другие возможности использования

Использование функциональности ориентируемого инструментального суппорта имеет смысл и тогда, когда на станке не предусмотрено физической возможности вращения инструментов, но жестко установлены инструменты с различной ориентацией. В этом случае измерение инструментов может осуществляться унифицировано в первичной ориентации, а релевантные для обработки размеры получаются через вращения виртуального инструментального суппорта.

11.3.1 Отражение длин инструмента

При установочных данных SD42900 \$SC_MIRROR_TOOL_LENGTH и SD42910 \$SC_MIRROR_TOOL_WEAR, отличных от нуля, возможно отражение компонентов длин инструмента и компонентов базовых размеров со значениями износа их соответствующих осей.



SD42900 \$SC_MIRROR_TOOL_LENGTH

Установочные данные **отличны** от нуля:

Компоненты длин инструмента (\$TC_DP3, \$TC_DP4 и \$TC_DP5) и компоненты базовых размеров (\$TC_DP21, \$TC_DP22 и \$TC_DP23), соответствующие оси которых отражены, также отражаются - через инверсию знака.

Значения износа при этом **не** отражаются. Если они также должны быть отражены, то должны быть установлены установочные данные SD42910 \$SC_MIRROR_TOOL_WEAR.

SD42910 \$SC_MIRROR_TOOL_WEAR

Установочные данные **отличны** от нуля:

Значения износа компонентов длин инструмента, соответствующие оси которых отражены, также отражаются - через инверсию знака.

11.3.2 Нормирование знака износа

При установочных данных SD42920 \$SC_WEAR_SIGN_CUTPOS и SD42930 \$SC_WEAR_SIGN, отличных от нуля, возможна инверсия нормирования знака компонентов износа.

SD42920 \$SC_WEAR_SIGN_CUTPOS

Установочные данные **отличны** от нуля:

У инструментов с релевантным положением резцов (токарные и шлифовальные инструменты, типы инструмента 400) нормирование знака компонентов износа в плоскости обработки зависит от положения резцов. Для типов инструментов без релевантного положения резцов эти установочные данные не имеют значения.

В таблице ниже через X обозначены размеры, знак которых инвертируется через SD42920 (отличны от 0):

Положение резца	длина 1	длина 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Примечание

Нормирования знака через SD42920 и SD42910 не зависят друг от друга. Если, к примеру, знак указания размера изменяется через оба установочных данных, то результирующий знак остается без изменений.

SD42930 \$SC_WEAR_SIGN

Установочные данные **отличны** от нуля:

Знак всех размеров износа инвертируется. Это действует как на длину инструмента, так и на прочие величины, как то радиус инструмента, радиус закругления и т.д.

Если вводится положительный размер износа, то тем самым инструмент становится "короче" и "тоньше", см. главу "Коррекция инструмента, специальная обработка", активация измененных установочных данных".

11.3.3 Система координат активной обработки (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

В зависимости от кинематики станка или от наличия ориентируемого инструментального суппорта измеренные в одной из этих систем координат значения износа переводятся или преобразуются в подходящую систему координат.

Системы координат активной обработки

Из следующих систем координат могут происходить смещения длины инструмента, которые включают компонент длины инструмента "износ" через соответствующую G-команду группы 56 в вычисление активного инструмента.

- Система координат станка (MCS)
- Базовая кинематическая система (BKS)
- Система координат детали (WCS)

- Система координат инструмента (TCS)
- Система координат инструмента кинематической трансформации (KCS)

Синтаксис

TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS

Объяснение

TOWSTD:	Значение по умолчанию для коррекций в длине инструмента, значение износа
TOWMCS:	Коррекции в длине инструмента в MCS
TOWWCS:	Коррекции в длине инструмента в WCS
TOWBCS:	Коррекции в длине инструмента в BKS
TOWTCS:	Коррекции длины инструмента на исходной точке инструментального суппорта (ориентируемый инструментальный суппорт)
TOWKCS:	Коррекции длины инструмента инструментальной головки (кинематической трансформации)

Дополнительная информация

Отличительные особенности

В следующей таблице представлены важнейшие отличительные особенности:

G-команда	Значение износа	Активный ориентируемый инструментальный суппорт
TOWSTD:	Установка по умолчанию, длина инструмента	Значения износа подвержены вращению.
TOWMCS:	Значение износа в MCS. TOWMCS идентична TOWSTD, если нет активного ориентируемого инструментального суппорта.	Вращается только вектор результирующей длины инструмента без учета износа.
TOWWCS:	Значение износа пересчитывается из WCS в MCS.	Вектор инструмента вычисляется без учета износа, как при TOWMCS.
TOWBCS:	Значение износа пересчитывается из BKS в MCS.	Вектор инструмента вычисляется без учета износа, как при TOWMCS.
TOWTCS:	Значение износа пересчитывается из системы координат инструмента в MCS.	Вектор инструмента вычисляется без учета износа, как при TOWMCS.

TOWWCS, TOWBCS, TOWTCS: вектор износа прибавляется к вектору инструмента.

Линейная трансформация

Длину инструмента можно рационально определить в MCS, только если MCS следует из BKS через линейную трансформацию.

Не линейная трансформация

Если, например, с TRANSMIT активна не линейная трансформация, то при указании MCS в качестве желаемой системы координат автоматически используется BKS.

Кинематическая трансформация отсутствует, ориентируемый инструментальный суппорт отсутствует

При отсутствии активной кинематической трансформации и ориентируемого инструментального суппорта все системы координат, кроме WCS, совпадают. Тем самым только WCS отличается от других систем координат. Поскольку нормируются только длины инструмента, смещения между системами координат не имеют значения.

Литература:

Дополнительная информация по коррекции инструмента представлена в Описании функций "Основные функции"; Коррекция инструмента (W1)

Учет значений износа

Установочные данные **SD42935 \$SC_WEAR_TRANSFORM** определяют, какой из трех компонентов износа:

- Износ
- Суммарные коррекции точные
- Суммарные коррекции грубые

необходимо подвергнуть вращению через трансформацию адаптера или ориентируемый инструментальный суппорт, если активна одна из следующих G-команд:

- TOWSTD
Установка по умолчанию. Для коррекций в длине инструмента.
- TOWMCS
Значения износа в системе координат станка (MCS).
- TOWWCS
Значения износа в системе координат детали (WCS).
- TOWBCS
Значения износа в базовой кинематической системе (BKS).

- TOWTCS
Значения износа в системе координат инструмента на зажиме инструмента (Т исходная точка инструментального суппорта).
- TOWKCS
Значения износа в системе координат инструментальной головки при кинематической трансформации.

Примечание

Нормирование отдельных компонентов износа (согласование с гео-осями, нормирование знака) управляется следующими факторами:

- Активная плоскость
- Трансформация адаптера
- Установочные данные:
 - SD42910 \$SC_MIRROR_TOOL_WEAR
 - SD42920 \$SC_WEAR_SIGN_CUTPOS
 - SD42930 \$SC_WEAR_SIGN
 - SD42940 \$SC_TOOL_LENGTH_CONST
 - SD42950 \$SC_TOOL_LENGTH_TYPE

11.3.4 Длина инструмента и смена плоскостей

При установочных данных SD42940 \$SC_TOOL_LENGTH_CONST, отличных от нуля, можно согласовать такие компоненты длин инструмента, как длина, износ и базовый размер, с геометрическими осями для токарных и шлифовальных инструментов при смене плоскости.

SD42940 \$SC_TOOL_LENGTH_CONST

Установочные данные **отличны** от нуля:

Согласование компонентов длин инструмента (длина, износ и базовый размер) с гео-осями при смене плоскости обработки (G17 - G19) не изменяется.

Таблица ниже показывает согласование компонентов длин инструмента с гео-осями для токарных и шлифовальных инструментов (тип инструмента 400 до 599):

Содержание	длина 1	длина 2	длина 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Любое значение, отличное от 0, которое не равно одному из шести приведенных значений, нормируется как значение 18.

Таблица ниже показывает согласование компонентов длин инструмента с гео-осями для всех других инструментов (тип инструмента < 400 или > 599):

Плоскость обработки	длина 1	длина 2	длина 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Любое значение, отличное от 0, которое не равно одному из шести приведенных значений, нормируется как значение 17.

Примечание

При представлении в таблицах предполагается, что гео-оси до 3 обозначаются как X, Y, Z. Для согласования коррекции с осью определяющим является не идентификатор оси, а последовательность осей.

11.4 Коррекция инструмента Online

11.4.1 Определение функции полинома (FCTDEF)

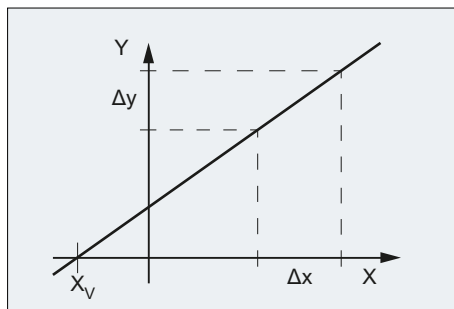
Определенные стратегии правки (к примеру, шарошка) характеризуются тем, что радиус шлифовального круга непрерывно (линейно) уменьшается с подачей шарошки. Для этого необходима линейная функция между подачей шарошки и записью значения износа соответствующей длины. Линейная функция определяется через предопределенную процедуру FCTDEF(...) для функций полинома макс. до третьей степени.

Уравнение прямой

$$y = f(x) = a_0 + a_1 \cdot x_1$$

a_1 : Подъем прямой, где $a_1 = \Delta x / \Delta y$

a_0 : Смещение прямой вдоль оси X, где $a_0 = -a_1 \cdot X_v$



Синтаксис

FCTDEF (<Func>, <LLimit>, <ULimit>, <a0>, <a1>, <a2>, <a3>)

Значение

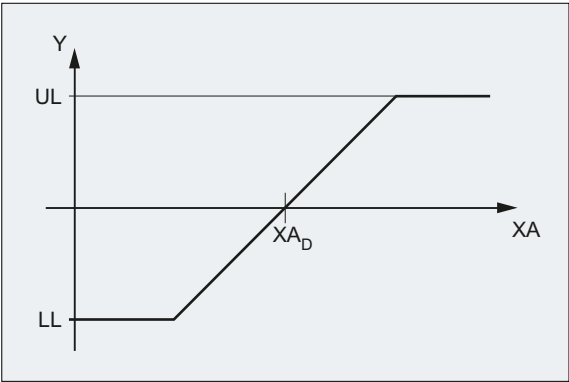
FCTDEF (...):	Определение функции полинома для PUTFTOCF(...): $y = f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$	
<Func>:	Номер функции	
	Тип данных:	INT
	Диапазон значений:	1, 2, 3
<LLimit>:	Нижняя величина ограничения	
	Тип данных:	REAL
<ULimit>:	Верхняя величина ограничения	
	Тип данных:	REAL
<a0>, <a1>, <a2>, <a3>:	Коэффициенты полиномиальной функции	
	Тип данных:	REAL

Пример

Определения

- Номер функции: 1
- Нижняя и верхняя величина ограничения: -100, 100
- Подъем характеристики: $a_1 = 1$
- Рабочая точка должна находиться в середине характеристики. Для этого характеристика должна быть смещена в отрицательном Y-направлении в соответствии с номинальной позицией оси XA в СКД к моменту определения функции в программе ЧПУ: $a_0 = -a_1 * XA_D = -1 * \AA_IW
- $a_2 = a_3 = 0$

Характеристика



- UL Верхняя величина ограничения
- LL Нижняя величина ограничения
- XA_D Заданное значение оси XA к моменту определения функции в программе ЧПУ

Программирование

Программный код	Комментарий
<code>FCTDEF (1, -100, 100, -\$AA_IW[XA], 1)</code>	; определение функций

11.4.2 Онлайн-запись коррекции инструмента, непрерывная (PUTFTOCF)

Предварительно определенная процедура PUTFTOCF(...) осуществляет онлайн-коррекцию инструмента при помощи предварительно определенной через FCTDEF(...) (Страница 443) функции полинома.

Примечание

Онлайн-коррекция инструмента может быть также осуществлена через синхронное действие.

Дополнительную информацию см. Справочник по функциям, синхронное действие.

Синтаксис

```
PUTFTOCF (<Func>, <RefVal>, <ToolPar>, <Chan>, <Sp>)
```

Значение

PUTFTOCF (...):	Запись онлайн-коррекции инструмента, непрерывно покадрово на основании функции полинома, определенной через FCTDEF(...)	
<Func>:	Номер функции, заданный при определении функции через FCTDEF(...)	
	Тип данных:	INT
	Диапазон значений:	1, 2, 3
<RefVal>:	Исходное значение, из которого должна выводиться коррекция (например, . заданное значение оси)	
	Тип данных:	VAR REAL
<ToolPar>:	Номер параметра износа (длина 1, 2 или 3), в котором должна быть учтена поправка.	
	Тип данных:	INT
<Chan>:	Номер канала, в котором должна действовать онлайн-коррекция инструмента	
	Примечание: Требуется только в том случае, если коррекция должна действовать не в активном канале.	
	Тип данных:	INT
<Sp>:	Номер шпинделя, для которого должна действовать онлайн-коррекция инструмента	
	Примечание: Требуется только в том случае, если вместо активного используемого инструмента должен быть исправлен неактивный шлифовальный круг.	
	Тип данных:	INT

11.4.3 Дискретная запись онлайн-коррекции инструмента (PUTFTOC)**Функция**

С помощью предварительно определенной процедуры PUTFTOC(...) онлайн-коррекция инструмента осуществляется с фиксированной поправкой.

Синтаксис

```
PUTFTOC (<CorrVal>, <ToolPar>, <Chan>, <Sp>)
```

Значение

PUTFTOC (...):	Запись онлайн-коррекции инструмента	
<CorrVal>:	Поправка, добавляемая к параметру износа	
	Тип данных:	REAL

<ToolPar>:	Номер параметра износа (длина 1, 2 или 3), в котором должна быть учтена поправка.	
	Тип данных:	INT
<Chan>:	Номер канала, в котором должна действовать онлайн-коррекция инструмента Примечание: Требуется только в том случае, если коррекция должна действовать не в активном канале.	
	Тип данных:	INT
<Sp>:	Номер шпинделя, для которого должна действовать онлайн-коррекция инструмента Примечание: Требуется только в том случае, если вместо активного используемого инструмента должен быть исправлен неактивный шлифовальный круг.	
	Тип данных:	INT

11.4.4 Включение/выключение онлайн-коррекции инструмента (FTOCON/FTOCOF)

Команды G FTOCON и FTOCOF включают / выключают онлайн-коррекцию инструмента.

Синтаксис

```
FTOCON
...
FTOCOF
```

Значение

FTOCON:	Включение онлайн-коррекции инструмента Команда должна быть запрограммирована в канале, в котором должна быть включена онлайн-коррекция инструмента.
FTOCOF:	Выключение онлайн-коррекции инструмента Команда должна быть запрограммирована в канале, в котором должна быть выключена онлайн-коррекция инструмента. Примечание: FTOCOF более не выводит коррекцию инструмента. В специфичных для резца данных коррекции сохраняется значение, рассчитанное в PUTFTOC/PUTFTOCF. Для окончательной деактивации онлайн-коррекции инструмента, после FTOCOF необходимо выполнить и включение/выключение инструмента (T...).

11.5 Коррекция радиуса инструмента 3D

11.5.1 Коррекция на радиус инструмента 3D для периферийного фрезерования 3D (CUT3DC, CUT3DCD, ISD)

Коррекция на радиус инструмента 3D (3D-КПИ) для периферийного фрезерования 3D без учёта ограничивающих поверхностей выбирается посредством действующей модально G-команды CUT3DC или CUT3DCD.

Сама активация выполняется посредством G41 или G42. Коррекция на радиус инструмента выключается посредством G40.

Синтаксис

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

Объяснение

CUT3DC:	3D-КПИ для периферийного фрезерования (только при активной 5-осевой трансформации)	
CUT3DCD:	3D-КПИ относительно дифференциального инструмента для периферийного фрезерования (только при активной 5-осевой трансформации) Разница радиусов устанавливается посредством параметра инструмента \$TC_DP15.	
G41/G42 X... Y... Z...:	Включение коррекции на радиус инструмента	
	G41:	Коррекция на радиус инструмента слева от контура
	G42:	Коррекция на радиус инструмента справа от контура
	Указание: Включение должно происходить в линейном кадре (G0/G1).	
CDOF2:	Выключение контроля столкновения для периферийного фрезерования 3D	
ORIC/ORID:	G-команды ORIC и ORID устанавливают поведение при изменениях ориентации на наружных углах.	
	ORIC:	Изменения ориентации на наружных углах накладываются на вставляемый круговой кадр.
	ORID:	Изменения ориентации на наружных углах выполняются перед вставляемым круговым кадром.

11.5 Коррекция радиуса инструмента 3D

ISD=<Value>:	Адрес ISD позволяет менять глубину врезания инструмента при периферийном фрезеровании и активной коррекции на радиус инструмента 3D.	
	<Value>:	Длина глубины врезания
G40 X... Y... Z...:	Выключение коррекции на радиус инструмента Указание: Выключение должно происходить в линейном кадре (G0/G1) с движениями гео-оси.	

Примечание

G-команды для выбора 3D-КПИ обрабатываются в кадре подвода, т. е. обычно в кадре, содержащем G41 или G42.

G41 или G42 могут быть запрограммированы и в кадрах без перемещения в значимых для коррекции гео-осях. В этом случае кадром подвода является первый кадр перемещения, следующий за таким кадром.

Смена варианта 3D-КПИ при активной коррекции на радиус инструмента игнорируется без аварийного сообщения.

Пример

Программный код	Комментарий
	; определение инструмента D1:
\$TC_DP1[1,1]=120	; тип (концевая фреза)
\$TC_DP3[1,1]=20	; вектор коррекции на длину
\$TC_DP6[1,1]=8	; радиус
N10 X0 Y0 Z0 T1 D1 F12000	; выбор инструмента.
N20 TRAORI(1)	; включение трансформации.
N30 G42 ORIC ISD=10 CUT3DC G64 X30	; активация периферийного фрезерования 3D, ; изменения ориентации на наружных углах (непрерывно), ; глубина врезания: 10 мм
N40 ORIWS A30 B15	; изменение ориентации на углу посредством указания позиций осей.
N50 Y20 A3=1 C3=1	; кадр перемещения с изменением ориентации, ; указание ориентации с вектором направления.
N60 X50 Y30	; кадр перемещения с постоянной ориентацией.
N70 Y50 A3=0.5 B3=1 C3=5	; кадр перемещения с изменением ориентации.
N80 M63	; кадр без информации перемещения.
N90 X0 ISD=20	; кадр перемещения с изменением глубины врезания.
N100 G40 Y0	; деактивация коррекции на радиус инструмента.
N110 M30	

Дополнительная информация

Траектория и ориентация

Используемый здесь вариант периферийного фрезерования реализован посредством задания траектории (направляющей линии) и соответствующей ориентации. При таком типе обработки форма инструмента на траектории не имеет значения. Значение имеет только радиус на полюсе зацепления инструмента.

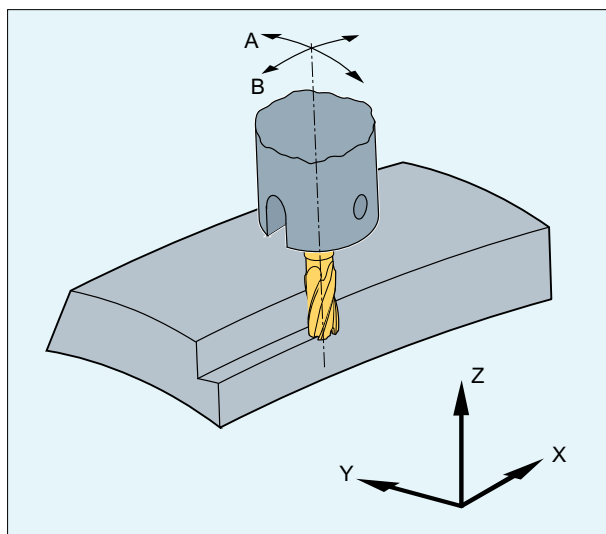


Рисунок 11-1 Периферийное фрезерование

Режим подвода

Режимом подвода в случае 3D-вариантов коррекции на радиус инструмента всегда является NORM.

Поведение на наружных углах

На наружных углах при периферийном фрезеровании с 3D-КПИ по аналогии с условиями при 2½D-КПИ обрабатываются G-команды группы 18 (поведение на углах коррекции на инструмент):

- G450: Переходная окружность (инструмент обходит углы детали по круговой траектории)
В отличие от решения при 2½D-КПИ добавленный элемент контура на наружном угле – это всегда окружность с радиусом 0, на которую коррекция на радиус инструмента воздействует так же, как и на любую другую запрограммированную траекторию. Вставка конических сечений вместо окружностей невозможна. Поэтому в этом случае адрес DISC не имеет значения и не обрабатывается.
- G451: Точка пересечения эквидистант (инструмент выполняет свободное резание в углу детали)
Для определения точки пересечения кривые смещения обоих задействованных кадров удлиняются, и определяется их точка пересечения в плоскости перпендикулярно к ориентации инструмента на углу.

Перемещение точки пересечения (G451) не используется, если между задействованными кадрами перемещения вставлен, по меньшей мере, один кадр,

содержащий изменение ориентации инструмента. В таком случае на углу всегда вставляется окружность.

Поведение при изменениях ориентации на наружных углах

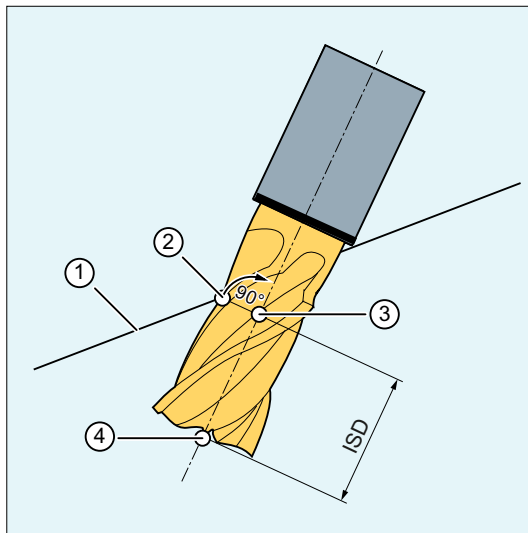
С помощью G-команд ORIC и ORID можно установить, будут ли изменения ориентации, запрограммированные между обоими образующими угол кадрами перемещения, выполняться перед началом вставленного кругового кадра (ORID) или одновременно с ним (ORIC).

Глубина врезания

Глубина врезания фрезы — это отступ вспомогательной точки фрезы от острия инструмента.

Вспомогательная точка фрезы — это вертикальная проекция точки обработки фрезы на запрограммированной траектории на продольную ось инструмента.

Так глубина врезания позволяет настраивать расположение точки обработки на боковой поверхности инструмента.



- ① Запрограммированная траектория
- ② Точка обработки фрезы
- ③ Вспомогательная точка фрезы
- ④ Острие фрезы

ISD Глубина врезания (InSertion Depth)

Рисунок 11-2 Глубина врезания

Коррекция радиуса инструмента относительно дифференциального инструмента

3D-КРИ относительно дифференциального инструмента для периферийного фрезерования активируется командой CUT3DCD. Ее следует применять, если контур запрограммирован относительно траектории центра стандартного инструмента, а обработка осуществляется другим инструментом. В этом случае при расчете коррекции на радиус инструмента 3D учитывают только значение износа радиуса активного инструмента (\$TC_DP15) и, при наличии, запрограммированные смещения коррекции

на инструмент OFFN и TOFFR. Базовый радиус (\$TC_DP6) активного инструмента **не** учитывается.

Фрезерование кармана с наклонными боковыми стенками для периферийного фрезерования посредством CUT3DC

При этой 3D-коррекции на радиус инструмента погрешность радиуса фрезы компенсируется посредством подачи в направлении нормалей к обрабатываемой поверхности. При этом плоскость, в которой лежит торцевая сторона фрезы, остается неизменной, если глубина врезания ISD не изменилась. Фреза с, например, меньшим радиусом по сравнению со стандартным инструментом, не достигла бы основания кармана, являющегося и ограничивающей поверхностью. Для автоматической подачи инструмента СЧПУ эта ограничивающая поверхность должна быть известна: см. главу "Периферийное фрезерование 3D с учетом ограничивающей поверхности (CUT3DCC, CUT3DCCD) (Страница 457)".

Advanced Surface / Top Surface

Примечание

При использовании коррекции на радиус инструмента CUT3DCD в сочетании с лицензируемой функцией "Advanced Surface" или "Top Surface" необходимо соблюдать рекомендации по настройке "Advanced Surface" / "Top Surface"!

Для проверки настроенных данных на портале SIOS имеются специальные программы проверки:

- программы проверки для Advanced Surface (<https://support.industry.siemens.com/cs/ww/en/view/78956392>)
- программы проверки для Top Surface (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)

11.5.2 Коррекция на радиус инструмента 3D для торцового фрезерования 3D (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

3D-коррекция на радиус инструмента (3D-КПИ) для торцового фрезерования 3D выбирается посредством G-команды (действует модально) CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD.

Сама активация выполняется посредством G41 или G42.

Для вычисления коррекции на радиус инструмента при торцовом фрезеровании 3D необходимо определение нормали к обрабатываемой плоскости. Она должна выполняться в кадре посредством G41 или G42 через адреса A4, B4, C4 и A5, B5, C5.

Коррекция на радиус инструмента выключается посредством G40.

Синтаксис

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
```

G40 X... Y... Z...

Объяснение

CUT3DFS:	3D-КРИ для торцового фрезерования с постоянной ориентацией. Ориентация инструмента установлена через G17 – G19 и не подвергается воздействию фреймов.	
CUT3DFF:	3D-КРИ для торцового фрезерования с постоянной ориентацией. Ориентация инструмента – это направление, установленное через G17 – G19 и, при необходимости, повернутое фреймом.	
CUT3DF:	3D-КРИ для торцового фрезерования с изменением ориентации (только при активной 5-осевой трансформации)	
CUT3DFD:	3D-КРИ относительно дифференциального инструмента для торцового фрезерования с изменением ориентации (только при активной 5-осевой трансформации) Разница радиусов устанавливается посредством параметра инструмента \$TC_DP15. Указание: CUT3DFD возможна только в сочетании с "Сглаживание нормалей к плоскости при торцовом фрезеровании 3D". Её активирует вызов лицензируемой функции "Top Surface" посредством CYCLE832(...) (Страница 825).	
G41/G42 X... Y... Z...:	Включение коррекции на инструмент Поведение при G41 и при G42 при торцовом фрезеровании 3D идентично. Указание: Включение должно происходить в линейном кадре (G0/G1).	
A4/5=... B4/5=... C4/5=...:	Определение нормали к обрабатываемой плоскости	
	A4=... B4=... C4=...:	Определение в начале кадра
	A5=... B5=... C5=...:	Определение в конце кадра
ORIC/ORID:	G-команды ORIC и ORID устанавливают поведение при изменениях ориентации на наружных углах.	
	ORIC:	Изменения ориентации на наружных углах накладываются на вставляемый круговой кадр.
	ORID:	Изменения ориентации выполняются перед круговым кадром.
G40 X... Y... Z...:	Выключение коррекции на радиус инструмента Указание: Выключение должно происходить в линейном кадре (G0/G1) с движениями гео-оси.	

Примечание

G41 или G42 могут быть запрограммированы и в кадрах без перемещения в значимых для коррекции гео-осях. В этом случае кадром подвода является первый кадр перемещения, следующий за таким кадром.

Смена варианта 3D-КПИ при активной коррекции на радиус инструмента игнорируется без аварийного сообщения.

Примеры**Пример 1: Торцовое фрезерование 3D посредством CUT3DF**

Программный код	Комментарий
N10	; определение инструмента D1:
N20 \$TC_DP1[1,1]=121	; тип инструмента (тороидальная фреза)
N30 \$TC_DP3[1,1]=20	; коррекция на длину
N40 \$TC_DP6[1,1]=5	; радиус
N50 \$TC_DP7[1,1]=3	; радиус закругления
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; выбор инструмента.
N90 TRAORI(1)	; выбор трансформации ориентации.
N100 B4=-1 C4=1	; определение плоскости.
N110 G41 ORID CUT3DF G64 X10 Y0 Z0	; включение коррекции на инструмент.
N120 X30	
N130 Y20 A4=1 C4=1	; наружный угол, переопределение плоскости.
N140 B3=1 C3=5	; изменение ориентации посредством ORID.
N150 B3=1 C3=1	; изменение ориентации посредством ORID.
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; изменение ориентации посредством ORIC
N180 A3=-1 C3=1	; изменение ориентации посредством ORIC
N190 Y-10 A4=-1 C4=3	; переопределение плоскости.
N200 X-20 Y-20 Z10	; внутренний угол с предшествующим кадром.
N210 X-30 Y10 A4=1 C4=1	; внутренний угол, переопределение плоскости.
N220 A3=1 B3=0.5 C3=1.7	; изменение ориентации с ORIC.
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; изменение ориентации.
N250 X0 Y30 C4=1	; движение по траектории, новая плоскость, ; ориентация с относительным программированием.
N260 BSPLINE X20 Z15	; начало сплайна, относительное программирование ориентации
N270 X30 Y25 Z18	; остается активным при сплайне.
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	

11.5 Коррекция радиуса инструмента 3D

Программный код	Комментарий
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; винтовая линия, ориентация с программированием осей.
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; движение по траектории, ориентация, не постоянная плоскость.
N330 X-20 G40	; отключение коррекции радиуса инструмента.
N340 M30	

Пример 2: Созданная из CAD-системы программа ЧПУ (фрагмент) с CUT3DFD

Программный код	Комментарий
N01 G710	
N03 T="12"	
N06 S5305 M03	
N07 G642	
; подвод к стартовой позиции в MCS с учётом длины инструмента.	
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
; конец позиционирования в MCS.	
;	
TRAORI(1)	; выбор трансформации ориентации.
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	; вызов CYCLE832 через: ; допуск контура = 0,01 мм, ; режим обработки: Top Surface со сглаживанием, ; чистовая обработка с вводом допуска ориентации, ; допуск ориентации = 1 градус
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; Кадры с N09 по N10 — быстрая доля движения подвода к детали с постоянной ориентацией.
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; В кадрах с N11 по N21 реализуется медленная доля движения подвода. Инструмент уже находится рядом с деталью, кроме того, теперь активны настройки изготовления пресс-форм (например, COMPSURF) из CYCLE832 (из-за активной G01). Путь траектории этой так называемой фазы входа для поведения изготовления пресс-форм должен составлять приблизительно в 1000 раз больше допуска контура (т. е., в этом примере: 10 мм).
N12 X-247.69126 Y33.82182 Z24.78219 F1061.00000	
N13 X-247.76560 Y33.13299 Z21.73022	
N14 X-247.82755 Y32.55897 Z19.18691	
N15 X-247.87918 Y32.08062 Z17.06748	
N16 X-247.92220 Y31.68200 Z15.30129	

Программный код	Комментарий
N17 X-247.95805 Y31.34981 Z13.82947 A3=-.216361686 B3=.934284391 C3=-.283373071	
N18 X-247.98792 Y31.07299 Z12.60295 A3=-.216360662 B3=.934280801 C3=-.283385691	
N19 X-248.01282 Y30.84230 Z11.58085 A3=-.216336015 B3=.934194446 C3=-.283689030	
N20 X-248.03357 Y30.65006 Z10.72910 A3=-.216233089 B3=.933833626 C3=-.284952647	
N21 X-248.05086 Y30.48986 Z10.01931 A5=-.060687572 B5=.974940255 C5=-.214029243 A3=-.215712821 B3=.932005189 C3=-.291263295	
N22 G41 X-248.06237 Y30.32400 Z9.36695 A5=-.060431854 B5=.973045457 C5=-.222554556 A3=-.214974689 B3=.929398552 C3=-.300007025 F1061.03295	; Начиная с кадра N22 нормаль к плоскости впервые полностью определена на протяжении всего кадра (т. е., нормаль к плоскости есть в конце предшествующего кадра N21 и, тем самым, в начале кадра N22 и в конце кадра N22). Таким образом выполнено условие включения коррекции на инструмент посредством G41/G42.
N23 X-248.07130 Y30.15119 Z8.71082 A5=-.060165696 B5=.971048883 C5=-.231179920 A3=-.214177198 B3=.926684940 C3=-.308841625	
N24 X-248.07829 Y29.97126 Z8.05094 A5=-.059884286 B5=.968941717 C5=-.239928784 A3=-.213318480 B3=.923853466 C3=-.317789237	
N25 X-248.08317 Y29.78487 Z7.38844 A5=-.059584206 B5=.966718449 C5=-.248807482 A3=-.212397895 B3=.920898045 C3=-.326854594	
N26 X-248.08578 Y29.59254 Z6.72679 A5=-.059263963 B5=.964380907 C5=-.257793037 A3=-.211418355 B3=.917822366 C3=-.336012474	
...	

Примечание

При торцовом фрезеровании 3D посредством CUT3DFD для включения коррекции на инструмент при помощи G41/G42 необходимо определение нормалей к плоскости. Программирование G41/G42 без определения нормалей к плоскости приводит к выведению аварийного сообщения.

Дополнительная информация

Торцовое фрезерование 3D

Для этого типа фрезерования 3D необходимо построчное описание траекторий 3D на поверхности детали. Обычно вычисления выполняются в CAM с учетом формы и размеров инструмента. Постпроцессор записывает в программу обработки детали — наряду с кадрами ЧПУ — ориентации инструмента (при активной 5-осевой трансформации) и G-команду желаемой 3D-коррекции на инструмент. Благодаря этому оператор станка может использовать немного меньшие инструменты, отличающиеся от инструмента, используемого для вычисления траекторий ЧПУ.

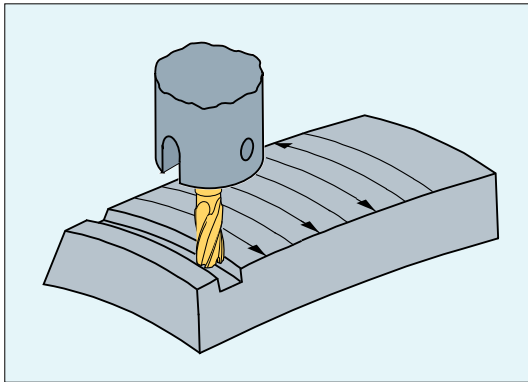


Рисунок 11-3 Торцовое фрезерование

Режим подвода

Режимом подвода в случае 3D-вариантов коррекции на радиус инструмента всегда является NORM.

Поведение на наружных углах

При торцовом фрезеровании наружные углы обрабатываются как окружности с радиусом 0, при этом плоскость окружности образуется конечной касательной первого и стартовой касательной второго кадра. Тем самым возможно изменение ориентации на переходе кадра. Поэтому на наружном угле окружность всегда вставляется как элемент контура. Перемещение точки пересечения при торцовом фрезеровании не доступно.

Поведение при изменениях ориентации на наружных углах

С помощью G-команд ORIC и ORID можно установить, будут ли изменения ориентации, запрограммированные между обоими образующими угол кадрами перемещения, выполняться перед началом вставленного кругового кадра (ORID) или одновременно с ним (ORIC).

Коррекция радиуса инструмента относительно дифференциального инструмента

3D-коррекция на радиус инструмента относительно дифференциального инструмента выбирается командой CUT3DFD. Ее следует применять, если контур запрограммирован относительно траектории центра стандартного инструмента, а обработка осуществляется другим инструментом. В этом случае при расчете коррекции на радиус инструмента 3D учитывают только значение износа радиуса активного инструмента (\$TC_DP15) и, при наличии, запрограммированные смещения коррекции на инструмент OFFN и TOFFR. Базовый радиус (\$TC_DP6) активного инструмента **не** учитывается.

Торцовое фрезерование 3D посредством CUT3DFD возможно только в сочетании с "Сглаживание нормалей к плоскости при торцовом фрезеровании 3D". Её активирует вызов лицензируемой функции "Top Surface" посредством CYCLE832(...) (Страница 825). Активацию необходимо выполнить **до** включения коррекции на инструмент посредством G41/G42, причём не непосредственно перед зацеплением инструмента, а заранее, за одну длину пути траектории, приблизительно равной 1000-кратному допуску контура (например, $1000 \times 0,01 \text{ мм} = 10 \text{ мм}$). Декрегирование должно выполняться в обратной последовательности: сначала выключение коррекции инструмента посредством G40, затем, после прохождения одной длины пути

траектории, соответствующей приблизительно 1000-кратному допуску контура, деактивация посредством, например, CUT2D (или подобной).

Чтобы иметь возможность применять "Сглаживание нормалей к плоскости при торцовом фрезеровании 3D", дополнительно необходимо разрешить функцию "Интерполяция нормалей к плоскости через полиномы":

MD28291 \$MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

Примечание

Для торцового фрезерования 3D посредством CUT3DFD в сочетании с функцией "Top Surface" необходимо соблюдать рекомендации по настройке в отношении "Top Surface"!

Для проверки настроенных данных на портале SIOS имеются (<https://support.industry.siemens.com/cs/ww/en/view/109738423>) специальные программы проверки.

11.5.3 Периферийное фрезерование 3D с учетом ограничивающей поверхности (CUT3DCC, CUT3DCCD)

При периферийном фрезеровании 3D с непрерывным или постоянным изменением ориентации инструмента часто программируется траектория центра инструмента для определенного стандартного инструмента. Поскольку на практике часто нет подходящих стандартных инструментов, можно использовать инструмент, не сильно отличающийся от стандартного ($\leq 5\%$).

CUT3DCCD позволяет учесть для реального дифференциального инструмента такую ограничивающую поверхность, которую описал бы запрограммированный стандартный инструмент. Программа ЧПУ описывает траекторию центра стандартного инструмента.

CUT3DCC учитывает при использовании цилиндрических инструментов такую ограничивающую поверхность, которой достиг бы запрограммированный стандартный инструмент. Программа ЧПУ описывает контур на поверхности обработки.

Вектор нормали к ограничивающей плоскости задается, как и при торцовом фрезеровании 3D, через A4, B4, C4 и A5, B5, C5.

Синтаксис

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

Объяснение

CUT3DCCD:	3D-КРИ для периферийного фрезерования с учетом ограничивающей поверхности с дифференциальным инструментом на траектории центра инструмента: Подача к ограничивающей поверхности	
CUT3DCC:	3D-КРИ для периферийного фрезерования с учетом ограничивающей поверхности с 3D-коррекцией радиуса: контур на поверхности обработки	
G41/G42 X... Y... Z...:	Включение коррекции на радиус инструмента	
	G41:	Коррекция на радиус инструмента слева от контура
	G42:	Коррекция на радиус инструмента справа от контура
	Указание: Включение должно происходить в линейном кадре (G0/G1).	
CDOF2:	Выключение контроля столкновения для периферийного фрезерования 3D	
A4/5=... B4/5=... C4/5=...:	Определение нормали к ограничивающей плоскости	
	A4=... B4=... C4=...:	Определение в начале кадра
	A5=... B5=... C5=...:	Определение в конце кадра
G40 X... Y... Z...:	Выключение коррекции на радиус инструмента	
	Указание: Выключение должно происходить в линейном кадре (G0/G1) с движениями гео-оси.	

Примечание

G-команды для выбора 3D-КРИ обрабатываются в кадре подвода, т. е. обычно в кадре, содержащем G41 или G42.

G41 или G42 могут быть запрограммированы и в кадрах без перемещения в значимых для коррекции гео-осях. В этом случае кадром подвода является первый кадр перемещения, следующий за таким кадром.

Смена варианта 3D-КРИ при активной коррекции на радиус инструмента игнорируется, аварийное сообщение не создается.

Пример

Программный код	Комментарий
N10 \$TC_DP1[1,1]=120	; цилиндрическая фреза
N20 \$TC_DP6[1,1]=10	
N30 \$TC_DP15[1,1]=-3	
...	
; обработка цилиндрической фрезой и CUT3DCCD	
N110 TRAORI	; включение трансформации.
N120 A4=0 B4=0 C4=1	; определение нормали к ограничивающей плоскости в начале кадра.
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000	

Программный код	Комментарий
N140 X10 Y0 Z0 G41 CUT3DCCD CDOF2 G64	; активация периферийного фрезерования 3D с учетом ограничивающей поверхности + выключение контроля столкновений.
N150 X20	
N160 X30 A45	; тупой угол ==> подача отсутствует
N170 X40 A-45	; острый угол ==> подача
N180 X55	
N190 Y10 Z10	; движение в направлении инструмента.
N200 Y20	
N210 C45	; чистое изменение ориентации.
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; изменение поверхности.
N240 Y50 G40	; деактивация коррекции на радиус инструмента.
...	

Дополнительная информация

Тип инструмента

Обрабатывается тип инструмента (параметр инструмента \$TC_DP1). Разрешены только фрезы с цилиндрическим хвостовиком (цилиндрическая или концевая фреза, тороидальная фреза и в граничном случае цилиндрическая фреза для изготовления штампов). Это соответствует типам инструментов с номерами 1 – 399, за исключением номеров 111, 155, 156, 157.

Стандартные инструменты с закруглением углов

Закругление углов стандартного инструмента описывается через параметр инструмента \$TC_DP7. По параметру инструмента \$TC_DP16 получается отклонение закругления углов реального инструмента от закругления углов стандартного инструмента.

Пример: Тороидальная фреза с радиусом меньше радиуса стандартного инструмента

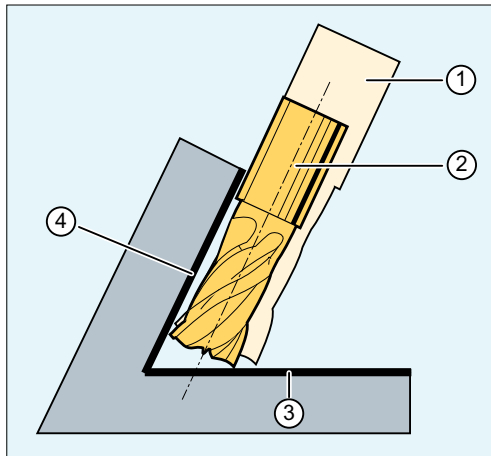
Тип инструмента	Радиус хвостовика (R)	Угловой радиус (r)
Стандартный инструмент с закруглением углов	$R = \$TC_DP6$	$r = \$TC_DP7$
Реальный инструмент с закруглением углов Типы инструмента 121 и 131 — тороидальная фреза (концевая фреза с закруглением углов)	$R' = \$TC_DP6 + \$TC_DP15 + OFFN$	$r' = \$TC_DP7 + \TC_DP16

В этом примере отрицательными являются и $\$TC_DP15 + OFFN$, и $\$TC_DP16$.

3D-КРИ через CUT3DCCD: Траектория центра инструмента с подачей до ограничивающей поверхности

Если используется инструмент, имеющий меньший радиус по сравнению с подходящим стандартным инструментом, то подаваемая в продольном направлении фреза

продолжает движение до тех пор, пока она снова не коснется основания кармана. Тем самым угол, образуемый поверхностью обработки и ограничивающей поверхностью, выбирается настолько, насколько это позволяет инструмент. При этом речь идет о смешанном принципе фрезерования (периферийного и торцового фрезерования). По аналогии с инструментом с уменьшенным радиусом в случае инструмента с увеличенным радиусом соответствующая подача выполняется в противоположном направлении.



- ① Стандартный инструмент
- ② Инструмент с меньшим радиусом, с подачей до ограничивающей поверхности
- ③ Ограничивающая поверхность
- ④ Поверхность обработки

В отличие от всех других коррекций на инструмент G-группы 22, заданный для CUT3DCCD параметр инструмента \$TC_DP6 не имеет значения для радиуса инструмента и не влияет на результирующую коррекцию. Смещение коррекции результирует из суммы значения износа радиуса инструмента (параметр инструмента \$TC_DP15) и смещения инструмента OFFN, запрограммированного для вычисления вертикального смещения к ограничивающей поверхности.

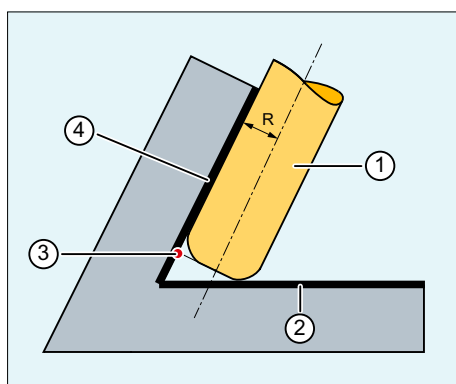
По созданной программы обработки детали не видно, лежит ли обрабатываемая поверхность слева или справа от траектории. Поэтому предполагается положительный радиус и отрицательное значение износа оригинального инструмента. Отрицательное значение износа всегда описывает инструмент с уменьшенным диаметром.

Использование цилиндрических инструментов

При использовании цилиндрических инструментов подача необходима, только если поверхность обработки и ограничивающая поверхность образуют острый угол (меньше 90 градусов). Если используется тороидальная фреза (концевая фреза с закруглением углов), то и для острых углов, и для тупых углов требуется подача в продольном направлении инструмента.

3D-КРИ через CUT3DCC: контур на поверхности обработки

Если CUT3DCC активна с тороидальной фрезой, то запрограммированная траектория относится к условной цилиндрической фрезе того же диаметра. Полученная таким образом исходная точка траектории при использовании тороидальной фрезы представлена на следующем рисунке.



- ① Тороидальная фреза
- ② Ограничивающая поверхность
- ③ Исходная точка траектории
- ④ Поверхность обработки
- R Радиус хвостовика (радиус инструмента)

Разрешен переход угла между поверхностью обработки и ограничивающей поверхностью из острого в тупой угол или наоборот и внутри одного кадра.

По сравнению со стандартным инструментом, используемый реальный инструмент может быть как больше, так и меньше. При этом результирующий угловой радиус не может стать отрицательным, и знак результирующего радиуса инструмента должен быть сохранен.

В случае CUT3DCC программа ЧПУ для обработки детали относится к контуру на обрабатываемой поверхности. При этом, как и при обычной коррекции на радиус инструмента, используется общий радиус, состоящий из следующих компонентов:

- Радиус инструмента (параметр инструмента \$TC_DP6),
- Значение износа (параметр инструмента \$TC_DP15),
- Смещение инструмента OFFN, запрограммированное для вычисления вертикального смещения к ограничивающей поверхности.

Положение ограничивающей поверхности определяется по следующей разнице:

Размеры стандартного инструмента – радиус инструмента (параметр инструмента \$TC_DP6)

Advanced Surface / Top Surface

Примечание

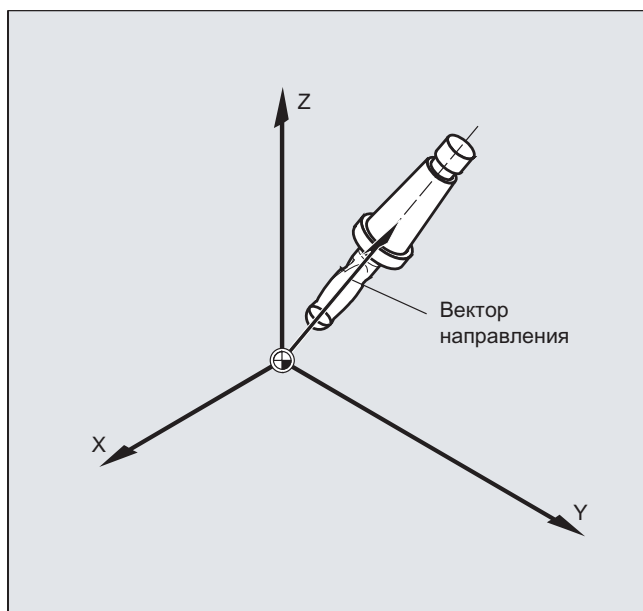
При использовании коррекции на радиус инструмента CUT3DCC / CUT3DCCD в сочетании с лицензируемой функцией "Advanced Surface" или "Top Surface" необходимо соблюдать рекомендации по настройке "Advanced Surface" / "Top Surface"!

Для проверки настроенных данных на портале SIOS имеются специальные программы проверки:

- программы проверки для Advanced Surface (<https://support.industry.siemens.com/cs/ww/en/view/78956392>)
 - программы проверки для Top Surface (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)
-

11.6 Ориентация инструмента (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

Под ориентацией инструмента понимается геометрическая точная установка инструмента в пространстве. Для 5-осевого станка ориентация инструмента может быть настроена через программные команды.



Активированные с OSD и OST движения перешлифовки ориентации, в зависимости от типа интерполяции, образуются для ориентации инструмента различно.

При активной векторной интерполяции сглаженная характеристика ориентации интерполируется и посредством векторной интерполяции. Напротив, при активной интерполяции круговой оси, ориентация сглаживается напрямую посредством движений круговых осей.

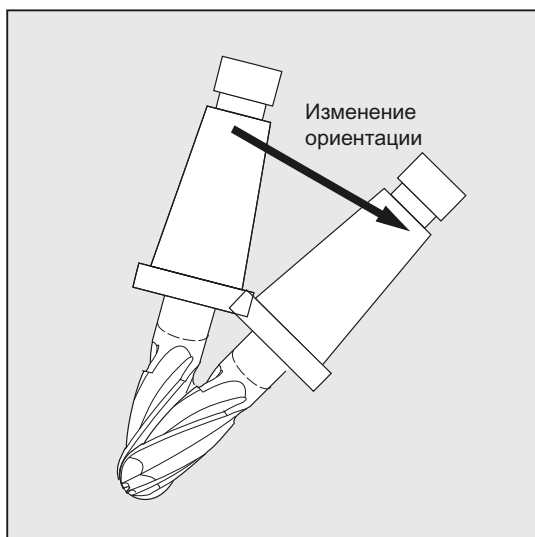
Программирование

Программирование изменения ориентации:

Изменение ориентации инструмента может быть запрограммировано через:

- Прямое программирование круговых осей A, B, C (интерполяция круговой оси)
- Угол Эйлера или угол RPY
- Вектор направления (векторная интерполяция через указание A3 или B3 или C3)
- LEAD/TILT (торцовое фрезерование)

Исходной системой координат является либо система координат станка (ORIMKS), либо текущая система координат детали (ORIWKs).

**Программирование ориентации инструмента:**

ORIC:	Ориентация и движение по траектории параллельно
ORID:	Ориентация и движение по траектории последовательно
OSOF:	Нет сглаживания ориентации
OSC:	Постоянная ориентация
OSS:	Сглаживание ориентации только в начале кадра
OSSE:	Сглаживание ориентации в начале и конце кадра
ORIS:	Скорость изменения ориентации при включенном сглаживании ориентации в градусах на мм (действует для OSS и OSSE)
OSD:	Перешлифовка ориентации через задачу длины перешлифовки с установочными данными: SD42674 \$SC_ORI_SMOOTH_DIST
OST:	Перешлифовка ориентации через задачу углового допуска в градусах при векторной интерполяции с установочными данными: SD42676 \$SC_ORI_SMOOTH_TOL При интерполяции круговой оси заданный допуск принимается как макс. погрешность осей ориентации.

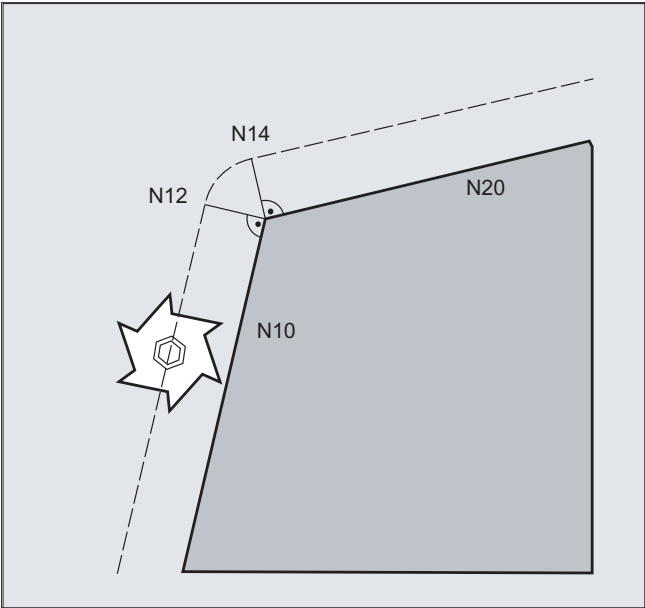
Примечание

Все команды по перешлифовке ориентации инструмента (OSOF, OSC, OSS, OSSE, OSD, OST) собраны в группе G-функций 34. Они действуют модально, т. е. одновременно может действовать только одна из этих команд.

Примеры

Пример 1: ORIC

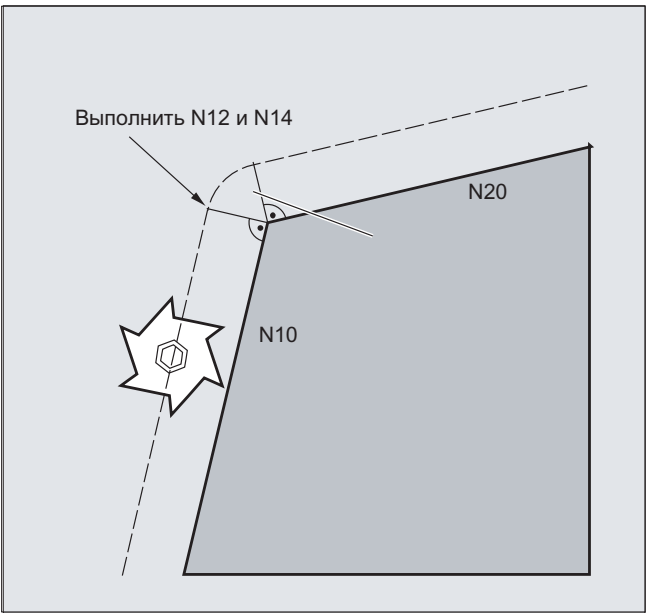
Если между кадрами перемещения N10 и N20 запрограммировано два или более кадров с изменениями ориентации (например, A2=... B2=... C2=...) и ORIC активна, то вставленный круговой кадр согласно величине угловых изменений распределяется на эти промежуточные кадры.



Программный код	Комментарий
ORIC	
N8 A2=... B2=... C2=...	
N10 X... Y... Z...	
N12 C2=... B2=...	; Круговой кадр, вставляемый на наружном углу, распределяется на N12 и N14, согласно изменению ориентации. Круговое движение и изменение ориентации при этом выполняются параллельно.
N14 C2=... B2=...	
N20 X =...Y=... Z=... G1 F200	

Пример 2: ORID

Если активна ORID, то все кадры между обоими кадрами перемещения выполняются в конце первого кадра перемещения. Круговой кадр с постоянной ориентацией выполняется непосредственно перед вторым кадром перемещения.

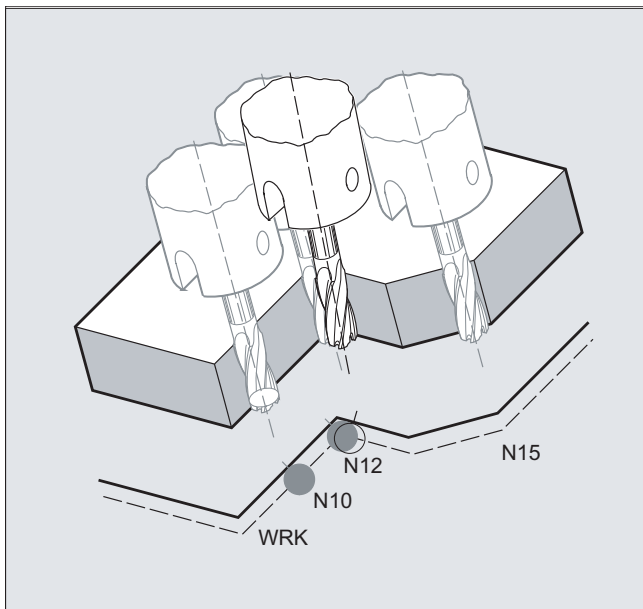


Программный код	Комментарий
ORID	
N8 A2=... B2=... C2=...	
N10 X... Y... Z...	
N12 A2=... B2=... C2=...	; Кадр N12 и N14 выполняется в конце N10. После выводится круговой кадр с текущей ориентацией.
N14 M20	; Вспомогательные функции и т. п.
N20 X... Y... Z...	

Примечание

Для типа изменения ориентации на наружном углу главной является программная команда, активная в первом кадре перемещения наружного угла.

Без изменения ориентации: Если ориентация на границе кадра не изменяется, то поперечным сечением инструмента является окружность, касающаяся обеих контуров.

Пример 3: Изменение ориентации на внутреннем угле**Программный код**

```

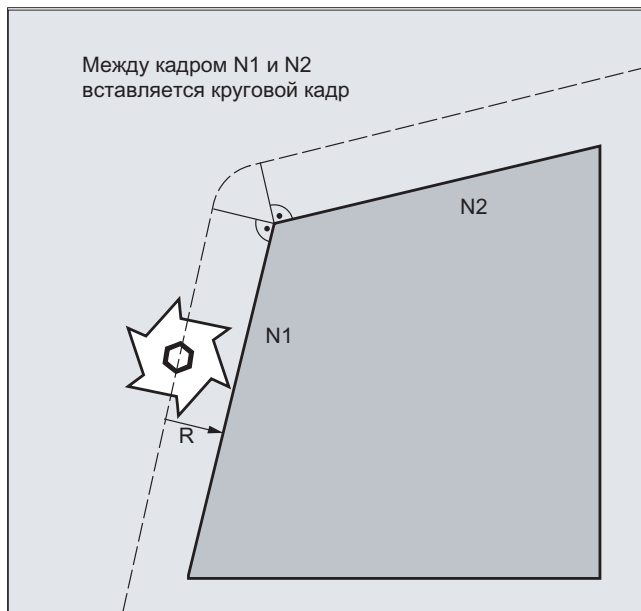
ORIC
N10 X ...Y... Z... G1 F500
N12 X ...Y... Z... A2=... B2=... C2=...
N15 X ...Y... Z... A2=... B2=... C2=...

```

Дополнительная информация**Поведение на наружных углах**

На наружном углу всегда вставляется круговой кадр с радиусом фрезы.

С помощью программных команд ORIC или ORID можно установить, будут ли изменения ориентации, запрограммированные между кадром N1 и N2, выполняться до начала вставленного кругового кадра или одновременно с ним.



Если на наружных углах необходимо изменение ориентации, то оно возможно по выбору параллельно с интерполяцией, или отдельно с движением по траектории.

При ORID сначала выполняются вставленные кадры без движения по траектории. Круговой кадр вставляется непосредственно перед вторым из двух кадров перемещения, образующих угол.

Если на наружном углу вставлено несколько кадров ориентации и ORIC выбрана, то круговое движение согласно значениям отдельных вставленных кадров распределяется на эти кадры.

Перешлифовка ориентации с OSD или OST

При перешлифовке с G642 макс. погрешности для контурных осей и осей ориентации не должны слишком сильно отличаться. Меньший из двух допуск определяет форму движения перешлифовки или угловой допуск, относительно сильное сглаживание характеристики ориентации, не принимая при этом больших погрешностей контура.

Посредством активации OSD или OST с заданной длиной перешлифовки или угловым допуском можно "щедро" сглаживать очень маленькие погрешности характеристики траектории без существенных погрешностей контура.

Примечание

В отличие от перешлифовки контура (и характеристики ориентации) с G642, при перешлифовке ориентации с OSD или OST отдельный кадр не образуется, а движение перешлифовки вставляется напрямую в запрограммированные оригинальные кадры.

С помощью OSD или OST невозможна перешлифовка переходов кадра, на которых выполняется смена типа интерполяции для ориентации инструмента (вектор → круговая ось, круговая ось → вектор). При необходимости перешлифовка этих переходов кадра возможна посредством обычных функций перешлифовки G641, G642 или G643.

11.7 Присвоение свободных номеров D, номеров резцов

11.7.1 Присвоение свободных номеров D, номеров резцов (адрес CE)

Номер D

Номера D могут использоваться в качестве номеров коррекции. Дополнительно через адрес CE возможна адресация номера резца. Через системную переменную \$TC_DPCE возможно описание номера резца.

Предустановка: № коррекции == номер резца

Через машинные данные устанавливается макс. число номеров D (номеров резцов) и макс. число резцов на инструмент (→ изготовитель станка). Следующие команды имеют смысл только тогда, когда макс. номер резца (MD18105) был установлен большим, чем число резцов на инструмент (MD18106). Следовать указаниям изготовителя станка.

Примечание

Наряду с присвоением относительных номеров D, номера D могут присваиваться и как "плоские" или "абсолютные" номера D (1-32000) без связи с номером T (внутри функции "Структура плоских номеров D").

Литература

Описание функций "Основные функции"; Коррекция инструмента (W1)

11.7.2 Присвоение свободных номеров D: Проверка номеров D (CHKDNO)

С помощью команды `CHKDNO` проверяется, присвоены ли имеющиеся номера D однозначно. Номера D всех определенных внутри блока TO инструментов могут встречаться только один раз. Запасные инструменты при этом не учитываются.

Синтаксис

```
state=CHKDNO (Tno1, Tno2, Dno)
```

Значение

state:	=TRUE:	Номера D были присвоены для проверенного диапазона однозначно.
	=FALSE:	Произошел конфликт номеров D или недействительное параметрирование. Через Tno1, Tno2 и Dno передаются параметры, приведшие к конфликту. Эти данные могут быть проанализированы в программе обработки детали.

CHKDNO (Tno1, Tno2):	Проверяются все номера D названных инструментов.
CHKDNO (Tno1):	Проверяются все номера D из Tno1 по сравнению со всеми остальными инструментами.
CHKDNO:	Проверяются все номера D всех инструментов по сравнению со всеми остальными инструментами.

11.7.3 Присвоение свободных номеров D: Переименование номеров D (GETDNO, SETDNO)

Номера D должны присваиваться однозначно. Два различных резца одного инструмента не могут иметь один и тот же номер D.

GETDNO

Эта команда выводит номер D определенного резца (ce) инструмента с номером T t. Если номер D для введенных параметров не существует, то устанавливается d=0. Если номер D является недействительным, то возвращается значение больше 32000.

SETDNO

С помощью этой команды присваивается значение d номера D резца ce инструмента t. Через state результат этого оператора возвращается (TRUE или FALSE). Если блок данных для введенных параметров не существует, то возвращается FALSE. Синтаксические ошибки приводят к сигнализации ошибки. Номер D не может быть явно установлен на 0.

Синтаксис

```
d = GETDNO (t, ce)
state = SETDNO (t, ce, d)
```

Значение

d:	Номер D резца инструмента
T:	Номер T инструмента
ce:	Номер резца (номер CE) инструмента
state:	Указывает, была ли команда выполнена без ошибок (TRUE или FALSE).

Пример переименования номера D

Программирование	Комментарий
\$TC_DP2[1,2]=120	
\$TC_DP3[1,2]=5.5	
\$TC_DPCE[1,2] = 3	;номер резца CE
...	
N10 def int DNrAlt, DNrNeu = 17	

Программирование	Комментарий
N20 DnrAlt = GETDNO(1,3)	
N30 SETDNO(1,3,DnrNeu)	

Таким образом, резцу CE=3 присваивается новое значение D 17. Теперь обращение к данным этого резца выполняется через номер D 17; как через системные переменные, так и при программировании с адресом ЧПУ.

11.7.4 Присвоение свободных номеров D: Определение номера T для заданного номера D (GETACTTD)

С помощью GETACTTD для абсолютного номера D определяется соответствующий номер T. Проверка однозначности не выполняется. Если существует несколько одинаковых номеров D внутри одного блока ТО, то возвращается номер T первого найденного инструмента. При использовании "плоских" номеров D использование команды не имеет смысла, так как здесь всегда возвращается значение "1" (нет номера T в системе УД).

Синтаксис

```
status=GETACTTD(Tnr,Dnr)
```

Значение

Dnr:	Номер D, для которого должен быть найден номер T.	
Tnr:	Найденный номер T	
status:	Величина:	Значение:
	0	Номер T был найден. Tnr содержит значение номера T.
	-1	Номера T для указанного номера D не существует; Tnr=0.
	-2	Номер D не абсолютный. Tnr получает значение первого найденного инструмента, содержащего номер D со значением Dnr.
	-5	Функция не была выполнена по иной причине.

11.7.5 Присвоение свободных номеров D: Установка номеров D как недействительных (DZERO)

Команда DZERO служит для поддержки при переоснащении. Обозначенные таким образом блоки данных коррекции более не проверяются языковой командой CHKDNO. Для того, чтобы снова сделать их доступными, D-номер снова должен быть установлен с SETDNO.

Синтаксис

```
DZERO
```

Значение

DZERO:	Обозначает все номера D блока TO как недействительные.
--------	--

11.8 Кинематика инструментального суппорта

Условия

Инструментальный суппорт может выполнять ориентацию инструмента во всех возможных пространственных направлениях только в том случае, если

- Имеется две оси вращения V_1 и V_2 .
- Оси вращения располагаются вертикально друг на друге.
- Продольная ось инструмента располагается вертикально на второй оси вращения V_2 .

Дополнительно для станков, для которых должны быть возможны любые ориентации, существует следующее требование:

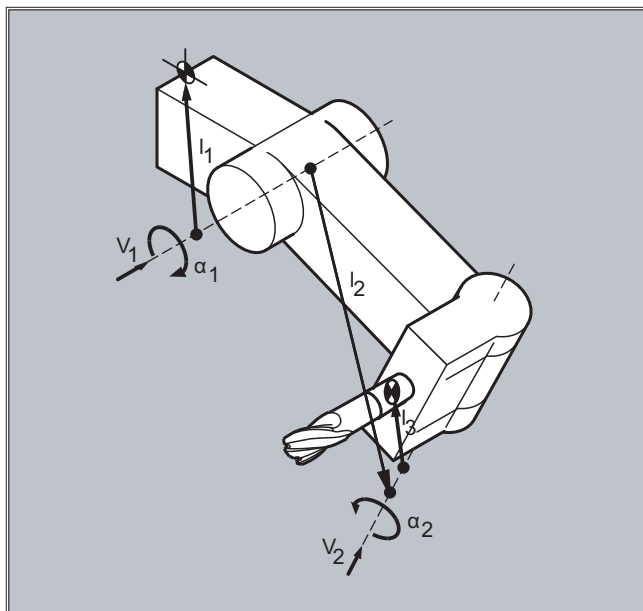
- Ориентация инструмента должна стоять вертикально на первой оси вращения V_1 .

Функция

Кинематика инструментального суппорта макс. с двумя осями вращения v_1 или v_2 описывается 17 системными переменными $\$TC_CARR1[m]$ до $\$TC_CARR17[m]$.

Описание инструментального суппорта состоит из:

- векторного расстояния от первой оси вращения до исходной точки инструментального суппорта I_1 , векторного расстояния от первой до второй оси вращения I_2 , векторного расстояния от второй оси вращения до исходной точки инструмента I_3 .
- векторов направления обеих осей вращения V_1 , V_2 .
- углов поворота α_1 , α_2 вокруг обеих осей. Углы поворота подсчитываются положительно при направлении взгляда в сторону векторов осей вращения по часовой стрелке.



Для станков с **подвижной кинематикой** (могут вращаться как инструмент, так и деталь) к системным переменным были добавлены элементы \$TC_CARR18[m] до \$TC_CARR23[m].

Параметры

Функция системных переменных для ориентируемых инструментальных суппортов			
Обозначение	Компонент x	Компонент y	Компонент z
I_1 вектор смещения	\$TC_CARR1[m]	\$TC_CARR2[m]	\$TC_CARR3[m]
I_2 вектор смещения	\$TC_CARR4[m]	\$TC_CARR5[m]	\$TC_CARR6[m]
v_1 ось вращения	\$TC_CARR7[m]	\$TC_CARR8[m]	\$TC_CARR9[m]
v_2 ось вращения	\$TC_CARR10[m]	\$TC_CARR11[m]	\$TC_CARR12[m]
α_1 угол поворота α_2 угол поворота	\$TC_CARR13[m] \$TC_CARR14[m]		
I_3 вектор смещения	\$TC_CARR15[m]	\$TC_CARR16[m]	\$TC_CARR17[m]

Расширения системных переменных для ориентируемых инструментальных суппортов			
Обозначение	Компонент x	Компонент y	Компонент z
I ₄ вектор смещения	\$TC_CARR18[m]	\$TC_CARR19[m]	\$TC_CARR20[m]
Идентификатор оси ось вращения v ₁ ось вращения v ₂	Идентификаторы осей вращения v ₁ и v ₂ (предустановка ноль) \$TC_CARR21[m] \$TC_CARR22[m]		
Тип кинематики	\$TC_CARR23[m]		
Tool Part Mixed mode	Тип кинематики-T ->	Тип кинематики-P ->	Тип кинематики-M
	Вращается только инструмент (предустановка).	Вращается только деталь	Вращаются деталь & инструмент
Смещение оси вращения v ₁ оси вращения v ₂	Угол в градусах осей вращения v ₁ и v ₂ при принятии первичной установки \$TC_CARR24[m] \$TC_CARR25[m]		
Угловое смещение оси вращения v ₁ оси вращения v ₂	Смещение торцового зубчатого зацепления в градусах осей вращения v ₁ и v ₂ \$TC_CARR26[m] \$TC_CARR27[m]		
Угловая разрешающая способность v ₁ ось вращения v ₂ ось вращения	Инкремент торцового зубчатого зацепления в градусах осей вращения v ₁ и v ₂ \$TC_CARR28[m] \$TC_CARR29[m]		
Мин. позиция ось вращения v ₁ ось вращения v ₂	Программный лимит для мин. позиции осей вращения v ₁ и v ₂ \$TC_CARR30[m] \$TC_CARR31[m]		

Расширения системных переменных для ориентируемых инструментальных суппортов			
Макс. позиция ось вращения v_1 ось вращения v_2	Программный лимит для макс. позиции осей вращения v_1 и v_2 \$TC_CARR32[m] \$TC_CARR33[m]		
Имя инструментального суппорта	Вместо цифры инструментальный суппорт может получить имя. \$TC_CARR34[m]		
Пользователь: имя оси 1 имя оси 2 идентификатор	Предполагаемое использование в измерительных циклах пользователя. \$TC_CARR35[m] \$TC_CARR36[m] \$TC_CARR37[m]		
Позиция	\$TC_CARR38[m]	\$TC_CARR39[m]	\$TC_CARR40[m]
Точное смещение	Параметры, которые могут быть прибавлены к значениям в базовых параметрах.		
l_1 вектор смещения	\$TC_CARR41[m]	\$TC_CARR42[m]	\$TC_CARR43[m]
l_2 вектор смещения	\$TC_CARR44[m]	\$TC_CARR45[m]	\$TC_CARR46[m]
l_3 вектор смещения	\$TC_CARR55[m]	\$TC_CARR56[m]	\$TC_CARR57[m]
l_4 вектор смещения	\$TC_CARR58[m]	\$TC_CARR59[m]	\$TC_CARR60[m]
v_1 ось вращения	\$TC_CARR64[m]		
v_2 ось вращения	\$TC_CARR65[m]		

Примечание**Пояснения к параметрам**

С "m" указывается соответствующий номер описываемого инструментального суппорта.

\$TC_CARR47 до \$TC_CARR54, а также \$TC_CARR61 до \$TC_CARR63, не определены и приводят к аварийному сообщению при попытке обращения по чтению или записи.

Начальные или конечные точки векторов расстояния на осях могут выбираться свободно. Углы поворота α_1 , α_2 вокруг обеих осей в исходном состоянии инструментального суппорта определяются с 0° . Таким образом, для описания кинематики инструментального суппорта имеется бесконечное число возможностей.

Инструментальные суппорта с только одной или без осей вращения могут быть описаны через установку на ноль векторов направления одной или обеих осей вращения. Для инструментального суппорта без оси вращения векторы расстояния действуют как дополнительные коррекции инструмента, компонент которых не затрагиваются переключением плоскостей обработки (G17 до G19)°.

Расширения параметров

Параметры осей вращения

Системные переменные были дополнены строками \$TC_CARR24[m] до \$TC_CARR33[m] и описаны следующим образом:

Смещение осей вращения v_1, v_2	Изменение позиции оси вращения v_1 или v_2 при первичной установке ориентируемого инструментального суппорта.
Угловое смещение/угловая разрешающая способность осей вращения v_1, v_2	Смещение или угловая разрешающая способность торцового зубчатого зацепления осей вращения v_1 и v_2 . Запрограммированный или вычисленный угол округляется до ближайшего значения, получаемого при целом n из $\phi = s + n * d$.
Мин. и макс. позиция оси вращения v_1, v_2	Мин. позиция/макс. позиция оси вращения, предельный угол (программный лимит) оси вращения v_1 и v_2 .

Параметры для пользователя

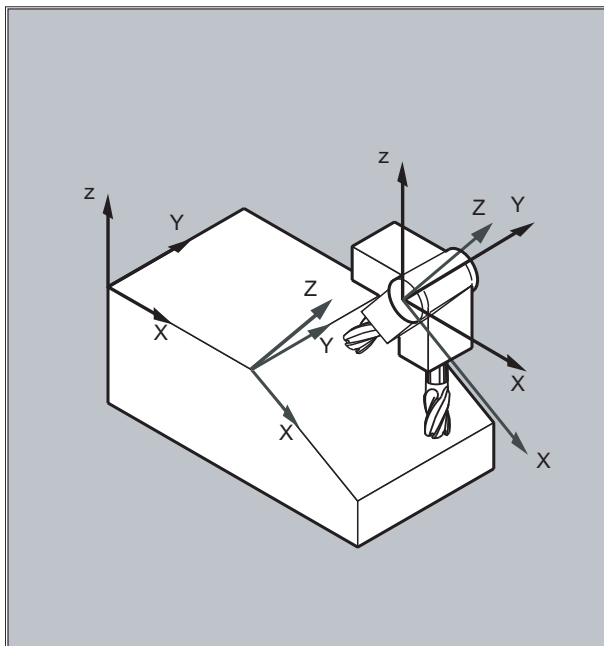
\$TC_CARR34 до \$TC_CARR40 содержат параметры, свободно доступные пользователю, которые до ПО 6.4 стандартно более не обрабатываются в NCK или не имеют значения.

Параметры точного смещения

\$TC_CARR41 до \$TC_CARR65 содержат параметры точного смещения, которые могут прибавляться к значениям в базовых параметрах. Согласованное с базовым параметром значение точного смещения получается через прибавление значения 40 к номеру параметра.

Пример

Используемый в следующем примере инструментальный суппорт может быть полностью описан через вращение вокруг оси Y.



Программный код	Комментарий
N10 \$TC_CARR8[1]=1	; Определение компонента Y первой оси вращения инструментального суппорта 1.
N20 \$TC_DP1[1,1]=120	; Определение концевой фрезы.
N30 \$TC_DP3[1,1]=20	; Определение концевой фрезы с длиной 20 мм.
N40 \$TC_DP6[1,1]=5	; Определение концевой фрезы с радиусом 5 мм.
N50 ROT Y37	; Определение фрейма с вращением в 37° вокруг оси Y.
N60 X0 Y0 Z0 F10000	; Подвод в исходную позицию.
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	; Установить коррекцию радиуса, коррекцию длин инструмента в повернутом фрейме, выбрать инструментальный суппорт 1, инструмент 1.
N80 X40	; Выполнить обработку при повороте на 37°.
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Дополнительная информация

Подвижная кинематика

Для станков с подвижной кинематикой (вращается как инструмент, так и деталь) к системным переменным были добавлены строки `$TC_CARR18 [m]` до `$TC_CARR23 [m]` и они были описаны следующим образом:

Поворотный инструментальный стол, состоящий из:

- векторного расстояния от второй оси вращения V_2 до исходной точки поворотного инструментального стола I_4 третьей оси вращения.

Круговые оси, состоящие из:

- обеих идентификаторов канала для исходной точки осей вращения V_1 и V_2 , к позиции которых при необходимости выполняется обращение при определении ориентации ориентируемого инструментального суппорта.

Тип кинематики с одним из значений T, P или M:

- Тип кинематики T: вращается только инструмент.
- Тип кинематики P: вращается только деталь.
- Тип кинематики M: вращаются инструмент и деталь.

Удаление данных инструментального суппорта

С помощью `$TC_CARR1 [0]=0` могут быть удалены данные всех блоков данных инструментального суппорта.

Типу кинематики `$TC_CARR23 [T]=T` должна быть присвоена одна из трех допустимых прописных или строчных букв (T,P,M) и по этой причине он не должен быть удален.

Изменение данных инструментального суппорта

Любое из описанных значений может быть изменено посредством присваивания нового значения в программе обработки детали. Любой другой символ, кроме T, P или M, приводит при попытке активации ориентируемого инструментального суппорта к ошибке.

Чтение данных инструментального суппорта

Любое из описанных значений может быть считано посредством присваивания переменной в программе обработки детали.

Точные смещения

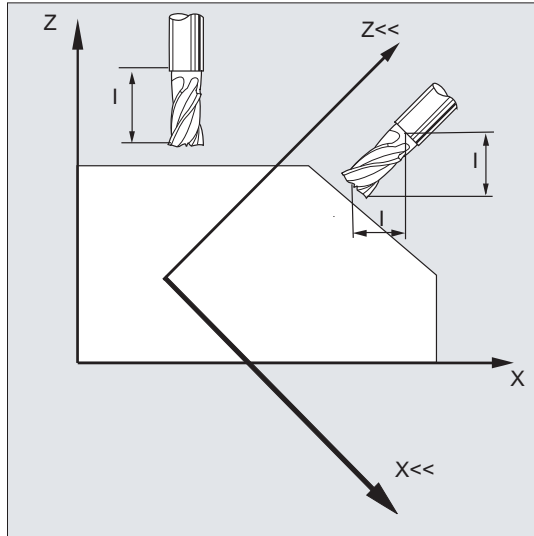
Недопустимое значение точного смещения определяется только тогда, когда активируется ориентируемый инструментальный суппорт, который содержит такое смещение и одновременно установочные данные SD42974

`$SC_TOCARR_FINE_CORRECTION = TRUE`

Величина допустимого точного смещения ограничивается через машинные данные до макс. допустимого значения.

11.9 Коррекция длин инструмента для ориентируемого инструментального суппорта (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

При изменении ориентации инструмента в пространстве изменяются и его компоненты длин.



Поэтому после переоснащения, к примеру, посредством ручной установки или замены инструментального суппорта с фиксированной точной установкой в пространстве, необходимо заново определить его компоненты длин инструмента. Это осуществляется с помощью команд перемещения TCOABS и TCOFR.

Для ориентируемого инструментального суппорта активного фрейма при выборе инструмента с TCOFRZ, TCOFRY и TCOFRX можно определить направление, в которое должен указывать инструмент.

Синтаксис

```
TCARR= [<m>]
TCOABS
TCOFR
TCOFRZ
TCOFRY
TCOFRX
```

Значение

TCARR= [<m>] :	Запросить инструментальный суппорт с номером "m"
TCOABS:	Определить компоненты длин инструмента из актуальной ориентации инструментального суппорта
TCOFR:	Определить компоненты длин инструмента из ориентации активного фрейма
TCOFRZ:	Ориентируемый инструментальный суппорт из активного фрейма, инструмент которого указывает в направлении Z

11.9 Коррекция длин инструмента для ориентируемого инструментального суппорта (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

TCOFRY:	Ориентируемый инструментальный суппорт из активного фрейма, инструмент которого указывает в направлении Y
TCOFRX:	Ориентируемый инструментальный суппорт из активного фрейма, инструмент которого указывает в направлении X

Дополнительная информация

Коррекция длин инструмента из ориентации суппорта (TCOABS)

TCOABS вычисляет коррекцию длин инструмента из актуальных углов ориентации инструментального суппорта; зафиксированы в системных переменных \$TC_CARR13 и \$TC_CARR14.

Для определения кинематики инструментального суппорта с системными переменными см. " Кинематика инструментального суппорта (Страница 473) ".

Для нового вычисления коррекции длин инструмента при смене фрейма необходимо заново выбрать инструмент.

Направление инструмента из активного фрейма

Ориентируемый инструментальный суппорт может быть установлен таким образом, что инструмент указывает в следующем направлении:

- с TCOFR или TCOFRZ в направлении Z
- с TCOFRY в направлении Y
- с TCOFRX в направлении X

Переключение между TCOFR и TCOABS вызывает новое вычисление коррекции длин инструмента.

Запросить инструментальный суппорт (TCARR)

С помощью TCARR через номер инструментального суппорта m запрашиваются его геометрические данные (память коррекций).

При m=0 активный инструментальный суппорт отключается.

Геометрические данные инструментального суппорта активируются только после вызова инструмента. Выбранный инструмент остается активным и после смены инструментального суппорта.

Актуальные геометрические данные инструментального суппорта могут быть определены и в программе обработки детали через соответствующие системные переменные.

Новое вычисление коррекции длин инструмента (TCOABS) при смене фрейма

Для нового вычисления коррекции длин инструмента при смене фрейма необходимо заново выбрать инструмент.

Примечание

Ориентация инструмента должна быть вручную согласована с активным фреймом.

11.9 Коррекция длин инструмента для ориентируемого инструментального суппорта (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

При вычислении коррекции длин инструмента на промежуточном этапе вычисляется и угол поворота инструментального суппорта. Т.к. у инструментальных суппортов с двумя осями вращения как правило существует две пары углов поворота, с помощью которых ориентация инструмента может быть согласована с активным фреймом, то зафиксированные в системных переменных значения углов поворота хотя бы приблизительно должны соответствовать установленным механически углам поворота.

Примечание

Ориентация инструмента

СЧПУ не может проверить вычисленный через ориентацию фрейма угол поворота на предмет возможности его установки на станке.

Если оси вращения инструментального суппорта конструктивно расположены таким образом, что вычисленная через ориентацию фрейма ориентация инструмента не может быть достигнута, то выводится ошибка.

Комбинация точной коррекции инструмента и функциональностей для коррекции длин инструмента при подвижных инструментальных суппортах запрещена. При попытке одновременного вызова обеих функций, следует сообщение об ошибке.

С помощью `TOFRAME` можно определить фрейм на основе направления ориентации выбранного инструментального суппорта. Подробнее см. в главе "Фреймы".

При активной трансформации ориентации (3-, 4-, 5-осевая трансформация) можно выбрать инструментальный суппорт с отличной от нулевого положения ориентацией, ошибка при этом не выводится.

Передаваемые параметры стандартных и измерительных циклов

Для передаваемых параметров стандартных и измерительных циклов действуют определенные диапазоны значений.

Для угловых значений диапазон значений определен следующим образом:

- Вращение вокруг 1-й гео-оси: от -180 до +180 градусов
- Вращение вокруг 2-й гео-оси: от -90 до +90 градусов
- Вращение вокруг 3-й гео-оси: от -180 до +180 градусов

См. главу "Фреймы", "Программируемое вращение (ROT, AROT, RPL)".

Примечание

При передаче угловых значений в стандартный или измерительный цикл учитывать:

Значения меньше дискретности вычисления ЧПУ округлять до нуля!

Дискретность вычисления ЧПУ для угловых позиций установлена в машинных данных:

`MD10210 $MN_INT_INCR_PER_DEG`

11.10 Коррекция длин инструмента Online (TOFFON, TOFFOF)

Через системную переменную \$AA_TOFF[<n>] возможно трехмерное наложение эффективных длин инструмента согласно трем направлениям инструмента в реальном времени.

В качестве индекса <n> используются три идентификатора гео-осей. Тем самым, число активных направлений коррекции определено через активные в это же время гео-оси.

Все коррекции могут быть активны одновременно.

Функция коррекции длин инструмента Online может использоваться для:

- Трансформация ориентации TRAORI
- Ориентируемого инструментального суппорта TCARR

Примечание

Коррекция длин инструмента Online это **опция**, требующая разрешения. Эта функция имеет смысл только в комбинации с активной трансформацией ориентации или активным ориентируемым инструментальным суппортом.

Синтаксис

```
TRAORI
TOFFON (<направление коррекции>[,<значение смещения>])
WHEN TRUE DO $AA_TOFF[<направление коррекции>]           ; В синхронных действиях.
...
TOFFOF (<направление коррекции>)
```

Прочие пояснения по программированию коррекции длин инструмента Online в синхронных действиях движения см. "Синхронные действия (Страница 641)".

Значение

TOFFON:	Активировать коррекцию длин инструмента Online	
	<направление коррекции>:	Направление инструмента (X, Y, Z), в котором должна действовать коррекция длин инструменте Online.
	<значение смещения>:	При активации для соответствующего направления коррекции может быть указано значение смещения, которое сразу же выводится.
TOFFOF:	Сбросить коррекцию длин инструмента Online Значения коррекции в указанном направлении коррекции сбрасываются и запускается остановка предварительной обработки.	

Примеры

Пример 1: Выбор коррекции длин инструмента

Программный код	Комментарий
MD21190 \$MC_TOFF_MODE =1	; Подвод к абсолютным значениям.
MD21194 \$MC_TOFF_VELO[0] =1000	
MD21196 \$MC_TOFF_VELO[1] =1000	
MD21194 \$MC_TOFF_VELO[2] =1000	
MD21196 \$MC_TOFF_ACCEL[0] =1	
MD21196 \$MC_TOFF_ACCEL[1] =1	
MD21196 \$MC_TOFF_ACCEL[2] =1	
N5 DEF REAL XOFFSET	
N10 TRAORI(1)	; Трансформация вкл.
N20 TOFFON(Z)	; Активация коррекции длин инструмента Online для направления инструмента Z.
N30 WHEN TRUE DO \$AA_TOFF[Z]=10 G4 F5	; Для направления инструмента Z выполняется интерполяция коррекции длин инструмента 10.
...	
N100 XOFFSET=\$AA_TOFF_VAL[X]	; Присвоить актуальную коррекцию в направлении X.
N120 TOFFON(X,-XOFFSET) G4 F5	; Для направления инструмента X коррекция длин инструмента снова возвращается на 0.

Пример 2: Отмена коррекции длин инструмента

Программный код	Комментарий
N10 TRAORI(1)	; Трансформация вкл.
N20 TOFFON(X)	; Активация коррекции длин инструмента Online для направления инструмента X.
N30 WHEN TRUE DO \$AA_TOFF[X] = 10 G4 F5	; Для направления инструмента X выполняется интерполяция коррекции длин инструмента 10.
...	
N80 TOFFOF(X)	; Смещение позиции направления инструмента X удаляется: ...\$AA_TOFF[X]=0 Оси не перемещаются. К актуальной позиции в WCS добавляется смещение позиции согласно актуальной ориентации.

Дополнительная информация

Подготовка кадра

При подготовке кадра на предварительной обработке действующее в главном ходе актуальное смещение длин инструмента также учитывается. Для возможности

практически полного использования макс. допустимых осевых скоростей необходимо остановить подготовку кадра с остановкой предварительной обработки `STOPRE` при установлении смещения инструмента.

Смещение инструмента всегда известно на момент предварительной обработки и тогда, когда коррекции длин инструмента после старта программы более не изменяются, или когда после изменения коррекций длин инструмента было обработано больше кадров, чем буфер IPO может принять между предварительной обработкой и главным ходом.

Переменная \$AA_TOFF_PREP_DIFF

Мера для разницы между актуальной действующей в интерполяторе коррекцией и коррекцией, которая действовала на момент подготовки кадра, может быть запрошена в переменной `$AA_TOFF_PREP_DIFF[<n>]`.

Установка машинных и установочных данных

Для коррекции длин инструмента Online имеются следующие системные данные:

- MD20610 \$MC_ADD_MOVE_ACCEL_RESERVE (резерв ускорения для наложенного движения)
- MD21190 \$MC_TOFF_MODE
Содержание системной переменной `$AA_TOFF[<n>]` выводится или интегрируется как абсолютное значение.
- MD21194 \$MC_TOFF_VELO (скорость коррекции длин инструмента Online)
- MD21196 \$MC_TOFF_ACCEL (ускорение коррекции длин инструмента Online)
- Установочные данные для задачи предельных значений:
SD42970 \$SC_TOFF_LIMIT (верхняя граница значения коррекции длин инструмента)

Литература:

Описание функций "Специальные функции"; F2 Многоосевые трансформации

11.11 Изменение данных коррекции для вращающихся инструментов

11.11.1 Вычисление ориентаций (ORISOLH)

Предварительно определенная функция ORISOLH помогает пользователю настраивать позиции круговых осей станка так, чтобы токарный инструмент приводился в определенное, не зависящее от кинематики положение относительно детали. В данном случае исходят из того, что активна 6-осевая трансформация, параметры которой заданы посредством кинематических цепей.

Доступны две основных функции:

- Точная установка инструмента
Задаются оба угла β и γ . Функция вычисляет необходимые для этого углы трёх осей ориентации.
- Точная установка инструмента (прямая)
Задаются углы второй и третьей осей ориентации. Функция вычисляет по ним соответствующие углы β и γ , а также угол недостающей первой оси ориентации.

Примечание

Последовательность осей ориентации

Если проходить кинематическую цепь, описывающую строение станка, от детали до инструмента, то для последовательности трёх осей ориентации 6-осевой трансформации действует следующее:

- Ближайшая к **детали** ось ориентации — **первая** ось ориентации.
- Ближайшая к **инструменту** ось ориентации — **третья** ось ориентации.

Первая ось ориентации, будет, как правило, шпинделем, и поэтому в таких случаях соответствующее вращение выполняется посредством вращающего фрейма.

Синтаксис

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

Объяснение

ORISOLH:	Вызов функции	
<RetVal>:	Возвращаемое значение функции	
	Тип данных:	INT
	Диапазон значений:	0, -2, -3, ..., -17
	Значения:	0 Функция была завершена без ошибок.
		-2 Ни одна из действительных трансформаций (6-осевая трансформация ориентации) не активна.
		-3 Первый параметр (<Cntrl>) отрицательный.
		-4 1-я позиция первого параметра (<Cntrl>) недействительна. Допустимы только значения 0 и 1.
		-5 10-я позиция первого параметра (<Cntrl>) недействительна. Допустимы только значения от 0 до 3.
		-6 100-я позиция первого параметра (<Cntrl>) недействительна. Допустимы только значения 0 и 1.
		-7 1000-я позиция первого параметра (<Cntrl>) недействительна. Допустимы только значения от 0 до 3.
		-8 В случае функции "Точная установка инструмента (прямая)" угол γ слишком большой.
		-9 В случае функции "Точная установка инструмента (прямая)", по меньшей мере, одна из заданных позиций осей нарушает границу оси.
		-10 Ни один из инструментов не активен.
		-11 Требуемая ориентация не может быть настроена.
		-12 Подгонка свободных осевых углов при хиртовом зубчатом зацеплении для первого или единственного решения невозможна.
		-13 Подгонка свободных осевых углов при хиртовом зубчатом зацеплении для второго решения невозможна.
		-14 Подгонка свободных осевых углов при хиртовом зубчатом зацеплении не возможна ни для одного из двух решений.
		-15 Первая ось ориентации параметризирована как хиртовая ось.
		-16 И вторая, и третья круговая ось параметризированы как хиртовые оси. Только одна из обеих осей может быть хиртовой.
		-17 В случае функции "Поворот прямо", по меньшей мере, одна из заданных позиций осей не совместима с соответствующим хиртовым зубчатым зацеплением.

11.11 Изменение данных коррекции для вращающихся инструментов

<Ctrl>:	управляет поведением функции	
	Тип данных:	INT
	Параметр <Ctrl> записан в десятичном коде (с 1-й позиции по 1000-ю позицию).	
	1-я позиция:	1-я позиция управляет поведением при ошибках:
	xxx0	В случае ошибки (возвращаемое значение < 0) обработка программы прерывается, и выводится аварийное сообщение 14106. Указание: Независимо от значения 1-й позиции аварийное сообщение выводится и в том случае, если параметр <Ctrl> отрицательный.
	xxx1	В случае ошибки (возвращаемое значение < 0) аварийное сообщение не выводится. Пользователь может самостоятельно надлежащим образом отреагировать на неполадку.
	10-я позиция:	управляет поведением при наличии оси ориентации с хиртовым зубчатым зацеплением: Указание: Этот параметр обрабатывается только в случае функции "Точная установка инструмента" (т. е., если 100-я позиция имеет значение "0").
	xx0x	Позиция оси округляется до ближайшей позиции.
	xx1x	Позиции осей округляются так, что отклонение угла β от его запрограммированного значения становится минимальным.
	xx2x	Позиции осей округляются так, что значение угла β равно максимально возможному значению, которое меньше запрограммированного значения (β округляется).
	xx3x	Позиции осей округляются так, что значение угла β равно минимально возможному значению, которое больше запрограммированного значения (β округляется).
	100-я позиция:	задает, какая функция должна быть выполнена, или какое значение имеют оба последующих параметра <W1> и <W2>.
	x0xx	Функция "Точная установка инструмента" Параметры <W1> и <W2> имеют следующее значение: • <W1> = β • <W2> = γ Вычисляются соответствующие углы осей ориентации.
	x1xx	Функция "Точная установка инструмента (прямая)" <W1> — заданное значение позиции второй оси ориентации, <W2> — заданное значение позиции третьей оси ориентации 6-осевой трансформации. Определяются позиция первой оси ориентации и углы β и γ , совместимые с обоими заданными значениями позиций. Если ошибка не появляется, в системной переменной \$P_ORI_POS[<n>, <m>] всегда выводятся два решения. При этом первый индекс <n> (0 или 1) ссылается на решение, а второй индекс <m> (0 ... 2) — на ось ориентации: • \$P_ORI_POS[0/1, 0]: Позиция первой оси ориентации • \$P_ORI_POS[0/1, 1]: Угол β

		• \$P_ORI_POS[0/1, 2]: Угол γ Проверяется, совместимы ли заданные значения позиций <W1> и <W2> с активными хиртовыми зубчатыми зацеплениями (если имеются) или активными программными лимитами. Если нет, возвращается соответствующий номер ошибки (см. параметр <RetVal>). При любом выборе обоих углов <W1> и <W2> резец инструмента, как правило, находится не в плоскости обработки. Угол γ, на который поворачивается резец из плоскости обработки, не должен быть больше предельного значения, определенного посредством установочных данных SD42999 \$SC_ORISOLH_INCLINE_TOL.
1000-я позиция:		Задаёт, как при необходимости модифицируются позиции решений, если 100-я позиция имеет значение "0", т. е. в случае функции "Точная установка инструмента".
	0xxx	Вычисленные позиции осей должны быть расположены максимально близко к актуальным полициям осей станка.
	1xxx	Вычисленные позиции осей должны быть в случае осей по модулю расположены максимально близко к центру диапазона по модулю, в случае других осей — максимально близко к 0. Для осей не по модулю это означает, что позиции осей ограничены диапазоном -180° ... +180°.
	2xxx	Вычисленные позиции осей должны быть независимо от типа оси ограничены диапазоном -180° ... +180°.
<W1>:	первый угол Значение результирует из 100-й позиции параметра <Cntrl>.	
	Тип данных:	REAL
<W2>:	Второй угол Значение результирует из 100-й позиции параметра <Cntrl>.	
	Тип данных:	REAL

Примечание

Незапрограммированные параметры имеют стандартное значение "0".

Дополнительная информация

Количество обнаруженных решений, а также другая информация о состоянии при выполнении функции ORISOLH можно считывать следующими системными переменными:

Системная переменная	Объяснение	
\$P_ORI_POS [<n>, <m>]	Возвращает углы осей ориентации, получающиеся при программировании ориентации.	
	<n>:	Индекс решения
	Диапазон значений:	0, 1
	<m>:	Индекс оси ориентации
	Диапазон значений:	0 ... 2 Последовательность осей ориентации (1 ... 3) относится к определению осей в \$NT_ROT_AX_NAME.
При вызове функции ORISOLH в режиме "Точная установка инструмента (прямая)" переменные \$P_ORI_POS[0/1, 1] и P_ORI_POS[0/1, 2] содержат относящиеся к обоим решениям значения обоих углов β и γ. Первое внесенное в \$P_ORI_POS[<n>, <m>] решение, т. е. с индексом <n> = 0, всегда то, которое выбирает система СЧПУ, если выполняется прямой подвод к требуемой ориентации. Второй индекс <m> относится при этом к оси ориентации, т. е. к \$NT_ROT_AX_NAME. Внесенные в \$P_ORI_POS[<n>, <m>] позиции осей учитывают смещения, внесенные в \$NK_OFF и \$NK_OFF_FINE, т. е., эти осевые углы могут использоваться в следующих кадрах без дальнейшей модификации для настройки желаемой ориентации. Если круговая ось является хиртовой осью, то позиции решения округляются до ближайшего следующего фиксированного положения хиртового зубчатого зацепления. Для круговых осей с хиртовым зубчатым зацеплением в системной переменной \$P_ORI_DIFF могут считываться разницы между позициями осей для точных решений и скорректированными по хиртовой развертке решениями.		
\$P_ORI_DIFF [<n>, <m>]	Возвращает разницу между точными и предоставляемыми в \$P_ORI_POS позициями осей ориентации, получающимися при программировании ориентации.	
	<n>:	Индекс решения
	Диапазон значений:	0, 1
	<m>:	Индекс оси ориентации
	Диапазон значений:	0 ... 2 Последовательность осей ориентации (1 ... 3) относится к определению осей в \$NT_ROT_AX_NAME.
Содержание может быть не равно нулю, только если позиции растрируются (хиртовое зубчатое зацепление), т. е., если системные данные \$NT_HIRTH_INCR соответствующей оси не равны нулю, и если эта ось является ручной круговой осью.		

Системная переменная	Объяснение	
\$P_ORI_SOL	Если при трансформации ориентации с более чем одной осью ориентации вычисляются осевые углы, которые должны вести к заданной ориентации, то существует более одного решения. В системной переменной \$P_ORI_SOL содержится количество действительных решений, а также дополнительная информация о состоянии. Содержание \$P_ORI_SOL кодировано следующим образом:	
	Значения < 0	Общие состояния ошибки
		-1 Для активной трансформации решения еще не были вычислены (отсутствие вызова ORISOLH).
		-2 Ни одна из трансформаций не активна, или активная трансформация не является трансформацией ориентации (6-осевая трансформация), которая может предоставить позиции для заданного программирования ориентации.
		-4 С имеющейся кинематикой желаемую ориентацию настроить невозможно.
		-5 При вызове функции ORISOLH в режиме "Точная установка инструмента (прямая)" решений обнаружено не было.
		-6 При вызове функции ORISOLH в режиме "Точная установка инструмента (прямая)" угол γ слишком большой.
		-7 При вызове функции ORISOLH в режиме "Точная установка инструмента (прямая)" был задан угол, который из-за хиртового зубчатого зацепления настроить невозможно.
		-8 Первая ось ориентации (фреймовая ось) не должна иметь параметры хиртовой оси.
		-9 И вторая, и третья круговая ось параметризованы как хиртовые оси. Только одна из обеих осей может быть хиртовой.
		-10 Подгонки решения (решений) по хиртовому зубчатому зацеплению обнаружено не было.
\$P_ORI_SOL	Значения > 0	Количество математически возможных решений без учёта границ оси и возможных условий ошибок.
	1-я позиция (разряд единиц)	0 Решения не существует, т. е., требуемую ориентацию настроить невозможно. У такого случая может быть три различных причины: - Требуемая ориентация принципиально недостижима из-за кинематики станка (не перпендикулярно размещенные оси ориентации) и при любом диапазоне перемещения осей ориентации. В этом случае 10-я и 100-я позиции \$P_ORI_SOL равняются нулю, а присвоенные осям ориентации переменные состояния \$P_ORI_STAT получают значение "-4". - Вычисленные решения недостижимы, поскольку они нарушили бы ограничения осей. Позиции осей ориентации, которые получились бы без ограничений осей, можно считать в \$P_ORI_POS. - При вызове функции ORISOLH в режиме "Точная установка инструмента (прямая)" позиции осей были

11.11 Изменение данных коррекции для вращающихся инструментов

Системная переменная	Объяснение		
			заданы так, что либо вектор ориентации, либо вектор ориентации нормали инструмента установлены параллельно первой оси ориентации, позицию которой необходимо вычислить. В этих случаях позиция этой оси не определена.
		1	Существует одно решение. У такого случая может быть три различных причины: • На основе заданной ориентации, кинематики станка и без учёта границ оси есть только одно решение (с математической точки зрения речь идёт о двух совпадающих решениях). Этот случай возникает в случае непрямоугольных кинематик на краю диапазона ориентации. \$P_ORI_POS содержит оба (идентичных) решения. • Есть только одно решение, поскольку из-за нарушения границ оси второе решение недействительно. В этом случае действительное решение — это всегда первое решение в \$P_ORI_POS. Второе решение, которое получилось бы без учёта границ оси, может быть также считано в \$P_ORI_POS. • При вызове функции ORISOLH в режиме "Точная установка инструмента (прямая)" это является обычной ситуацией. По заданным позициям двух осей ориентации существует только одна действительная позиция подлежащей вычисления отсутствующей оси ориентации.
		2	Существует два решения.
		8	Существует бесконечное количество решений, т. е., позиция оси ориентации (полярной оси) — любая. Однако из обоих возможных положений другой оси (других осей) одно исключено из-за нарушения границ оси.
		9	Существует бесконечное количество решений, т. е., позиция оси ориентации (полярной оси) — неопределённая. Неопределённую ось можно определить по 100-й позиции или по системной переменной \$P_ORI_STAT.
	Значения > 0 10-я позиция (разряд десятков)	Индикация (в двоичном коде) нарушенных границ оси. Точную причину ошибки можно определить по системной переменной \$P_ORI_STAT.	
		Бит 0 (значение 10):	По меньшей мере, для одного решения, нарушена, по меньшей мере, одна граница 1-й оси ориентации.
		Бит 1 (значение 20):	По меньшей мере, для одного решения, нарушена, по меньшей мере, одна граница 2-й оси ориентации.
		Бит 2 (значение 40):	По меньшей мере, для одного решения, нарушена, по меньшей мере, одна граница 3-й оси ориентации.

Системная переменная	Объяснение		
	Значения > 0 100-я позиция (разряд сотен)	Индикация (в двоичном коде) не определенных позиций осей (может возникать, только если есть бесконечное количество решений, т. е., если 1-я позиция равна "9").	
		Бит 0 (значение 100):	Позиция 1-й оси ориентации не определена.
		Бит 1 (значение 200):	Позиция 2-й оси ориентации не определена.
		Бит 2 (значение 400):	Позиция 3-й оси ориентации не определена.
	Обозначения 1-й, 2-й и 3-й оси ориентации относятся к определению осей в \$NT_ROT_AX_NAME.		

11.11 Изменение данных коррекции для вращающихся инструментов

Системная переменная	Объяснение	
\$P_ORI_STAT [<n>]	Возвращает для каждой из максимально 3 осей ориентации состояние после вызова ORISOLH.	
	<n>:	Индекс оси ориентации (соответствует индексу соответствующей оси ориентации в \$NT_ROT_AX_NAME)
		Диапазон значений: 0 ... 2
	Последовательность осей ориентации (1 ... 3) относится к определению осей в \$NT_ROT_AX_NAME.	
	Содержание \$P_ORI_STAT кодировано следующим образом:	
	Значения < 0	Общие состояния ошибки
		-1 Состояние не определено (отсутствие вызова ORISOLH).
		-2 Ни одна из трансформаций не активна, или активная трансформация не является трансформацией ориентации (6-осевая трансформация), которая может предоставить позиции для заданного программирования ориентации.
		-3 В активной трансформации ось не содержится.
		-4 Позицию оси вычислить невозможно, поскольку при имеющейся кинематике требуемая ориентация недостижима независимо от принятого диапазона перемещения оси.
		-5 При вызове функции ORISOLH в режиме "Точная установка инструмента (прямая)" позиции осей были заданы так, что либо вектор ориентации, либо вектор ориентации нормали инструмента установлены параллельно первой оси ориентации, позицию которой необходимо вычислить. В этих случаях позиция этой оси не определена.
		-6 При вызове функции ORISOLH в режиме "Точная установка инструмента (прямая)" угол γ слишком большой.
		-7 При вызове функции ORISOLH в режиме "Точная установка инструмента (прямая)" был задан угол, который из-за хиртового зубчатого зацепления настроить невозможно.
		-8 Первая ось ориентации (фреймовая ось) не должна иметь параметры хиртовой оси.
		-9 И вторая, и третья круговая ось параметризованы как хиртовые оси. Только одна из обеих осей может быть хиртовой.
		-10 Подгонки решения (решений) по хиртовому зубчатому зацеплению обнаружено не было.
	Значения > 0	Индикация (в двоичном коде) нарушенных границ оси первого решения.
	1-я позиция (разряд единиц)	Бит 0 (значение 1): Первое решение нарушает нижнюю границу оси.
		Бит 1 (значение 2): Первое решение нарушает верхнюю границу оси.
	Значения > 0	Индикация (в двоичном коде) нарушенных границ оси второго решения.
	10-я позиция (раз-	Бит 0 (значение 10): Второе решение нарушает нижнюю границу оси.

Системная переменная	Объяснение		
	ряд десятков)	Бит 1 (значение 20):	Второе решение нарушает верхнюю границу оси.
	Значения > 0	Индикация не определенной позиции оси.	
	100-я позиция (разряд сотен)	Бит 0 (значение 100):	Позиция оси ориентации не определена, т. е., требуемая ориентация достигается при любой настройке круговой оси (полюсное положение). Эта информация содержится также в системной переменной \$P_ORI_SOL.
	Одновременно может отображаться несколько номеров ошибок, указывающих на нарушение границ оси. Поскольку при нарушении границы оси предпринимается попытка достижения позиции в пределах разрешенных границ оси посредством сложения или вычитания кратного 360°, то (если это невозможно) нет однозначного определения того, которая граница оси была нарушена — верхняя или нижняя. Если для требуемой ориентации решения нет (\$P_ORI_SOL = 0), то состояние содержащихся в трансформации осей ориентации равно "0".		

Примечание**\$NT_ROT_AX_NAME**

Эта системная переменная ссылается не более чем на 3 оси, служащие для настройки ориентации. Она содержит имена элементов цепи (\$NK_NAME), определяющих те оси станка (круговые оси), которые должны выполнять движения ориентации, результирующих из кинематической трансформации. Последовательность, в которой содержится не более трех круговых осей в этой системной переменной, не имеет значения для кинематики станка, поскольку она выводится из структуры кинематических цепей. Поскольку она определяет последовательность, в которой другие переменные обращаются к круговым осям, последовательность осей ориентации в \$NT_ROT_AX_NAME должна совпадать с описанием кинематики.

Примечание**Информация о состоянии**

Информация о состояниях, которая отражает, например, что ориентация недостижима или достижима только с нарушением значимых границ оси, не вызывает аварийного сообщения ЧПУ. В обязанности пользователя входит должное реагирование на указанные условия.

11.11.2 Активация модификации данных коррекции для вращающихся инструментов (CUTMOD, CUTMODK)

Модификация данных коррекции для вращающихся инструментов активируется в программе ЧПУ языковой командой CUTMOD (в сочетании с ориентируемыми инструментальными суппортами) или CUTMODK (для трансформаций ориентации, определяемых посредством кинематических цепей).

Примечание

Поскольку ориентируемые инструментальные суппорты и трансформации ориентации, определенные посредством кинематических цепей, не могут быть активны одновременно, конфликтов между обоими вариантами не возникает.

Синтаксис

CUTMOD = <Value>

или

CUTMODK = <Command>

Объяснение

CUTMOD:	Вызов функции в сочетании с ориентируемыми инструментальными суппортами		
<Value>:	присвоенное значение		
	Тип данных:	INT	
	Значение:	0	Функция деактивирована. Возвращаемые системной переменной \$P_AD... значения идентичны соответствующим параметрам инструмента.
		> 0	Функция активируется, если активен ориентируемый инструментальный суппорт с указанным номером, т. е. активация привязана к определенному ориентируемому инструментальному суппорту. Возвращаемые системной переменной \$P_AD... значения возможно изменены по сравнению с соответствующими параметрами инструмента в зависимости от активного вращения. Деактивация обозначенного ориентируемого инструментального суппорта деактивирует функцию временно, активация другого ориентируемого инструментального суппорта деактивирует ее постоянно. Поэтому в первом случае функция снова активируется при повторном выборе того же ориентируемого инструментального суппорта, во втором случае необходим повторный выбор, причем и в том случае, когда позднее ориентируемый инструментальный суппорт с указанным номером активируется заново. Reset не влияет на функцию.
		-1	Функция активируется всегда при активном ориентируемом инструментальном суппорте. При смене инструментального суппорта или при его отключении и повторном выборе позднее, повторной установки CUTMOD не требуется.
		-2	Функция активируется всегда при активном ориентируемом инструментальном суппорте, номер которого идентичен такому актуального активного ориентируемого инструментального суппорта. Если нет активного ориентируемого инструментального суппорта, то это равнозначно CUTMOD=0. Если ориентируемый инструментальный суппорт активен, то это равнозначно прямому указанию актуального инструментального суппорта.
		< -2	Значения меньше 2 игнорируются, т. е. этот случай обрабатывается так, как если бы CUTMOD не была запрограммирована. Указание: Этот диапазон значений не должен использоваться, т.к. он зарезервирован для возможных будущих расширений.
CUTMODK:	Вызов функции в сочетании с трансформациями ориентации, определяемыми посредством кинематических цепей		

11.11 Изменение данных коррекции для вращающихся инструментов

<Command>:	присвоенная команда		
	Тип данных:	STRING	
	Значение:	"NEW"	Сохраняются значимые для функции "Модификация данных коррекции" состояния активной трансформации, определенной посредством кинематических цепей, имя трансформации и актуальный фрейм контура. Указание: Эта команда допустима, только если подходящая трансформация (TRAORI_DYN, TRAORI_STAT или TRAANG_K) активна.
		"OFF"	Выключает активную функцию "Модификация данных коррекции". Сохраненные до этого командой "NEW" данные сохраняются. Указание: Эта команда допустима также, если CUTMODK не активна. В этом случае она не оказывает влияния. Блок данных для "Модификация данных коррекции" (при наличии) сохраняется.
		"ON"	Эта команда повторно активирует "Модификацию данных коррекции" с блоком данных, сохраненным ранее командой "NEW". Если при выполнении этой команды активна трансформация с именем сохраненного набора данных, "Модификация данных коррекции" немедленно становится действительной. В противном случае активация откладывается до тех пор, пока не будет активирована текущая трансформация.
		"CLEAR"	Как и команда "OFF", выключает функцию "Модификация данных коррекции" и в дополнение к этому удаляет сохраненный блок данных. Указание: Эта команда допустима также, если CUTMODK не активна.

Примечание**SD42984 \$SC_CUTDIRMOD**

Активируемая командой CUTMOD или CUTMODK функция заменяет активируемую через установочные данные SD42984 \$SC_CUTDIRMOD функцию. Но эта функция продолжает оставаться доступной без изменений. Но поскольку параллельное использование обеих функций нерационально, она может быть активирована, только если CUTMOD равна нулю, а CUTMODK равна нулевой строке.

Дополнительная информация

Считывание модифицированных данных коррекции

Модифицированные данные коррекции предоставляются в следующих системных переменных и переменных BTSS:

Объяснение	Системная переменная	BTSS-переменная
Положение резца	\$P_AD[2]	cuttEdgeParam2
Угол державки	\$P_AD[10]	cuttEdgeParam10
Направление резания	\$P_AD[11]	cuttEdgeParam11
Задний угол	\$P_AD[24]	cuttEdgeParam24

Данные всегда модифицированы по сравнению с соответствующими параметрами инструмента (\$TC_DP2[...], ...), если функция "Модификация данных коррекции для вращающихся инструментов" была активирована командой CUTMOD или CUTMODK, и инструмент вращался посредством ориентируемого инструментального суппорта или подходящей трансформации ориентации.

Другие значимые для функций системные переменные

Системная переменная	Объяснение
\$P_CUTMOD_ANG / \$AC_CUTMOD_ANG	Возвращает угол, на который был повернут инструмент в активной плоскости обработки, и который лежит в основе определения модифицированных данных резцов функций CUTMOD или CUTMODK.
\$P_CUTMOD / \$AC_CUTMOD	Считывает текущее действительное значение, запрограммированное последним с помощью команды CUTMOD (номер инструментального суппорта, для которого необходимо активировать модификацию данных коррекции). Если последнее запрограммированное значение CUTMOD=-2 (активация с текущим активным ориентируемым инструментальным суппортом), то в системной переменной возвращается не значение "-2", а номер активного на момент программирования ориентируемого инструментального суппорта.
\$P_CUTMODK / \$AC_CUTMODK	Считывает имя трансформации, под которым был сохранён действительный на данный момент блок данных для "Модификация данных коррекции".

11.11 Изменение данных коррекции для вращающихся инструментов

Системная переменная	Объяснение
\$P_CUT_INV / \$AC_CUT_INV	Возвращает значение TRUE, если инструмент повернут таким образом, что необходима инверсия направления вращения шпинделя. Для этого в кадре, к которому относится соответствующая операция чтения, должны быть выполнены следующие четыре условия: 1. Активен токарный или шлифовальный инструмент (типы инструмента 400 до 599 и/или SD42950 \$SC_TOOL_LENGTH_TYPE = 2). 2. Модификация данных коррекции была активирована командой CUTMOD или CUTMODK. 3. Активны ориентируемый инструментальный суппорт или определенная посредством кинематических цепей трансформация ориентации, который/которая были выбраны командой CUTMOD или CUTMODK. 4. Ориентируемый инструментальный суппорт или кинематическая трансформация ориентации вращают инструмент таким образом, что результирующая нормаль резца инструмента повернута более чем на 90° по отношению к исходному положению (обычно 180°). Если, по меньшей мере, одно из названных условий не выполнено, то переменная возвращает значение FALSE. Для инструментов, положение резцов которых не определено, значение переменной всегда FALSE.

Системная переменная	Объяснение																		
\$P_CUTMOD_ERR	Ошибка после последнего вызова функции CUTMOD Функция CUTMOD может быть вызвана и не явно при смене инструмента. При сбросе (Reset) переменная сбрасывается на нуль. При каждой смене инструмента она сначала сбрасывается на нуль и при необходимости описывается заново. Переменная кодирована в битах. Биты имеют следующие значения: <table border="1"> <tr> <td>Бит 0:</td><td>Для активного инструмента действительное направление резания не определено.</td></tr> <tr> <td>Бит 1:</td><td>Оба угла резца (задний угол и угол державки) активного инструмента равны нулю.</td></tr> <tr> <td>Бит 2:</td><td>Задний угол активного инструмента имеет недопустимое значение ($< 0^\circ$ или $> 180^\circ$).</td></tr> <tr> <td>Бит 3:</td><td>Угол державки активного инструмента имеет недопустимое значение ($< 0^\circ$ или $> 90^\circ$).</td></tr> <tr> <td>Бит 4:</td><td>Угол режущей пластины активного инструмента имеет недопустимое значение ($< 0^\circ$ или $> 90^\circ$).</td></tr> <tr> <td>Бит 5:</td><td>Комбинация "положение резца – угол державки" активного инструмента недопустима (при положении резцов от 1 до 4 угол державки должен быть $\leq 90^\circ$, при положении резцов от 5 до 8 он должен быть $\geq 90^\circ$).</td></tr> <tr> <td>Бит 6:</td><td>Недопустимое вращение активного инструмента. Инструмент был вывернут на $\pm 90^\circ$ (с погрешностью приблизительно в 1°) из активной плоскости обработки. Из-за этого положение резца в плоскости обработки более не определено.</td></tr> <tr> <td>Бит 7:</td><td>Режущая пластина расположена не в плоскости обработки, и угол между режущей пластиной и плоскостью обработки превышает верхнюю границу, заданную в установочных данных SD42998 \$SC_CUTMOD_PLANE_TOL.</td></tr> <tr> <td>Бит 8:</td><td>Режущая пластина расположена не в плоскости обработки. Величина угла α больше 1°. Угол α — угол поворота вокруг оси координат, которая перпендикулярна как оси вращения угла β, так и оси вращения угла γ (для G18 — ось X).</td></tr> </table>	Бит 0:	Для активного инструмента действительное направление резания не определено.	Бит 1:	Оба угла резца (задний угол и угол державки) активного инструмента равны нулю.	Бит 2:	Задний угол активного инструмента имеет недопустимое значение ($< 0^\circ$ или $> 180^\circ$).	Бит 3:	Угол державки активного инструмента имеет недопустимое значение ($< 0^\circ$ или $> 90^\circ$).	Бит 4:	Угол режущей пластины активного инструмента имеет недопустимое значение ($< 0^\circ$ или $> 90^\circ$).	Бит 5:	Комбинация "положение резца – угол державки" активного инструмента недопустима (при положении резцов от 1 до 4 угол державки должен быть $\leq 90^\circ$, при положении резцов от 5 до 8 он должен быть $\geq 90^\circ$).	Бит 6:	Недопустимое вращение активного инструмента. Инструмент был вывернут на $\pm 90^\circ$ (с погрешностью приблизительно в 1°) из активной плоскости обработки. Из-за этого положение резца в плоскости обработки более не определено.	Бит 7:	Режущая пластина расположена не в плоскости обработки, и угол между режущей пластиной и плоскостью обработки превышает верхнюю границу, заданную в установочных данных SD42998 \$SC_CUTMOD_PLANE_TOL.	Бит 8:	Режущая пластина расположена не в плоскости обработки. Величина угла α больше 1° . Угол α — угол поворота вокруг оси координат, которая перпендикулярна как оси вращения угла β , так и оси вращения угла γ (для G18 — ось X).
Бит 0:	Для активного инструмента действительное направление резания не определено.																		
Бит 1:	Оба угла резца (задний угол и угол державки) активного инструмента равны нулю.																		
Бит 2:	Задний угол активного инструмента имеет недопустимое значение ($< 0^\circ$ или $> 180^\circ$).																		
Бит 3:	Угол державки активного инструмента имеет недопустимое значение ($< 0^\circ$ или $> 90^\circ$).																		
Бит 4:	Угол режущей пластины активного инструмента имеет недопустимое значение ($< 0^\circ$ или $> 90^\circ$).																		
Бит 5:	Комбинация "положение резца – угол державки" активного инструмента недопустима (при положении резцов от 1 до 4 угол державки должен быть $\leq 90^\circ$, при положении резцов от 5 до 8 он должен быть $\geq 90^\circ$).																		
Бит 6:	Недопустимое вращение активного инструмента. Инструмент был вывернут на $\pm 90^\circ$ (с погрешностью приблизительно в 1°) из активной плоскости обработки. Из-за этого положение резца в плоскости обработки более не определено.																		
Бит 7:	Режущая пластина расположена не в плоскости обработки, и угол между режущей пластиной и плоскостью обработки превышает верхнюю границу, заданную в установочных данных SD42998 \$SC_CUTMOD_PLANE_TOL.																		
Бит 8:	Режущая пластина расположена не в плоскости обработки. Величина угла α больше 1° . Угол α — угол поворота вокруг оси координат, которая перпендикулярна как оси вращения угла β , так и оси вращения угла γ (для G18 — ось X).																		

\$P_...: Переменные предварительной обработки

\$AC_...: Переменные главного хода

Все переменные главного хода могут считываться в синхронных действиях. Обращение по чтению из предварительной обработки создает останов предварительной обработки.

Смена плоскостей

Для определения измененного положения резца, направления резания и угла державки или заднего угла определяющим является рассмотрение резца в соответствующей активной плоскости (G17 – G19).

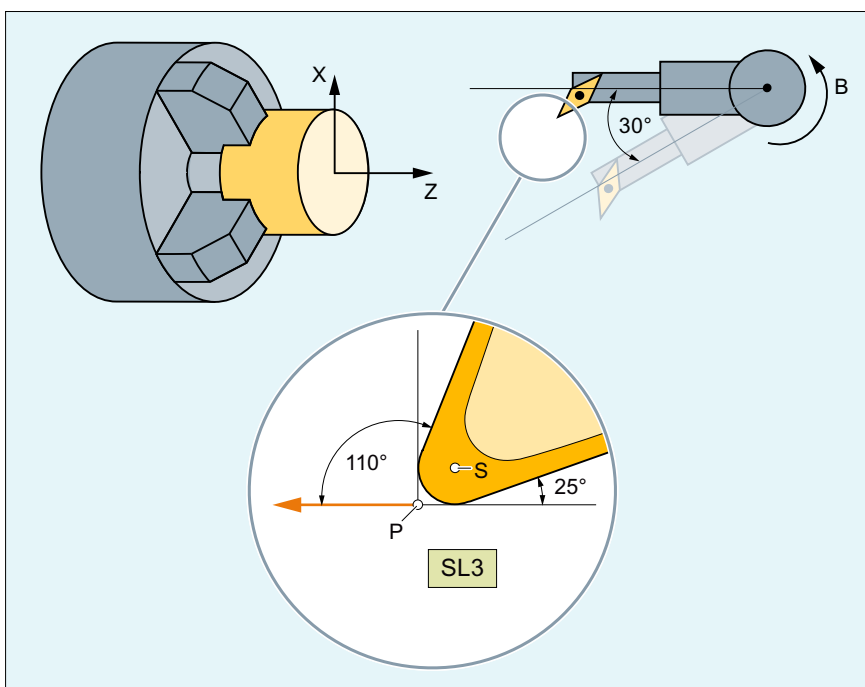
Если же установочные данные SD42940 \$SC_TOOL_LENGTH_CONST (смена компонентов длин инструмента при смене плоскостей) содержат действительное значение, отличное от нуля (плюс или минус 17, 18 или 19), то их содержание определяет плоскость, в которой рассматриваются релевантные величины.

Это правило преимущества установочных данных по отношению к G-кодам можно отменить, установив бит 18 машинных данных \$MC_TOOL_PARAMETER_DEF_MASK.

Это означает, что если этот бит установлен, то всегда действует плоскость, определенная посредством G-команды группы 6.

Активность измененных данных резцов

Измененное положение резцов и измененная опорная точка резцов сразу же активируются при программировании и для уже активного инструмента. Нового выбора инструмента для этого не требуется.



S: Центр резца

P: Исходная точка резца

SL: Положение резца

Рисунок 11-4 Инструмент с положением резца 3 и ориентируемым инструментальным суппортом, который может вращать инструмент вокруг оси В.

Программный код	Комментарий
N10 \$TC_DP1[1,1]=500	
N20 \$TC_DP2[1,1]=3	; Положение резцов
N30 \$TC_DP3[1,1]=12	
N40 \$TC_DP4[1,1]=1	
N50 \$TC_DP6[1,1]=6	
N60 \$TC_DP10[1,1]=110	; Угол державки
N70 \$TC_DP11[1,1]=3	; Направление резания
N80 \$TC_DP24[1,1]=25	; Задний угол
N90 \$TC_CARR7[2]=0 \$TC_CARR8[2]=1 \$TC_CARR9[2]=0	; Ось В
N100 \$TC_CARR10[2]=0 \$TC_CARR11[2]=0	; Ось С
\$TC_CARR12[2]=1	
N110 \$TC_CARR13[2]=0	

11.11 Изменение данных коррекции для вращающихся инструментов

Программный код	Комментарий
N120 \$TC_CARR14[2]=0	
N130 \$TC_CARR21[2]=X	
N140 \$TC_CARR22[2]=X	
N150 \$TC_CARR23[2]="M"	
N160 TCOABS CUTMOD=0	
N170 G18 T1 D1 TCARR=2	; X Y Z
N180 X0 Y0 Z0 F10000	; 12.000 0.000 1.000
N190 \$TC_CARR13[2]=30	
N200 TCARR=2	
N210 X0 Y0 Z0	; 10.892 0.000 -5.134
N220 G42 Z-10	; 8.696 0.000 -17.330
N230 Z-20	; 8.696 0.000 -21.330
N240 X10	; 12.696 0.000 -21.330
N250 G40 X20 Z0	; 30.892 0.000 -5.134
N260 CUTMOD=2 X0 Y0 Z0	; 8.696 0.000 -7.330
N270 G42 Z-10	; 8.696 0.000 -17.330
N280 Z-20	; 8.696 0.000 -21.330
N290 X10	; 12.696 0.000 -21.330
N300 G40 X20 Z0	; 28.696 0.000 -7.330
N310 M30	

Числовые значения в комментариях указывают соответствующие позиции конца кадра в координатах станка (MCS) в последовательности X → Y → Z.

Пояснения

В кадре N180 сначала выбирается инструмент при CUTMOD=0 и не повернутом ориентируемом инструментальном суппорте. Т.к. все векторы смещения ориентируемого инструментального суппорта 0, выполняется подвод к позиции, соответствующей указанным в \$TC_DP3[1,1] и \$TC_DP4[1,1] длинами инструмента.

В кадре N200 ориентируемый инструментальный суппорт активируется с вращением в 30° вокруг оси В. Т.к. положение резцов из-за CUTMOD=0 не изменяется, определяющей как и прежде является старая опорная точка резцов. Поэтому в кадре N210 выполняется подвод к позиции, содержащей старую опорную точку резцов в нулевой точке (т.е. вектор (1, 12) поворачивается в плоскости Z/X на 30°).

В кадре N260 действует в отличие от кадра N200 CUTMOD=2. Из-за вращения ориентируемого инструментального суппорта получается измененное положение резца 8. Следствием этого являются и отклонения в позициях осей.

В кадрах N220 или N270 активируется коррекция на радиус инструмента (КРИ). Различное положение резцов в обоих отрезках программы не влияет на конечные позиции кадров, в которых активна КРИ, поэтому соответствующие позиции идентичны. Только в кадрах сброса N260 или N300 различные положения резцов снова сказываются.

11.12 Работа с инструментальным окружением

Обзор функций

- Сохранение инструментального окружения (TOOLENV) (Страница 503)
- Удаление инструментального окружения (DELTOOLENV) (Страница 506)
- Чтение номеров T, D и DL (GETTENV) (Страница 507)
- Считывание компонентов длины инструмента или компонентов длины (GETTCOR) (Страница 509)
- Изменение компонентов инструмента (SETTCOR) (Страница 516)

Обзор системных переменных

- Считывание данных сохраняемого инструментального окружения (\$P_TOOLENVN, \$P_TOOLENV) (Страница 508)

11.12.1 Сохранение инструментального окружения (TOOLENV)

Функция TOOLENV служит для сохранения всех текущих состояний, имеющих значение для нормирования находящихся в памяти данных инструмента.

По отдельности это следующие данные:

- Активная G-команда группы:
 - 6 (G17, G18, G19)
 - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- Активная поперечная ось
- Машинные данные:
 - MD18112 \$MN_MM_KIND_OF_SUMCORR (свойства суммарных коррекций в области TO)
 - MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (определение параметров инструмента)

- Установочные данные:
 - SD42900 \$SC_MIRROR_TOOL_LENGTH (смена знака длины инструмента при отражении)
 - SD42910 \$SC_MIRROR_TOOL_WEAR (смена знака износа инструмента при отражении)
 - SD42920 \$SC_WEAR_SIGN_CUTPOS (знак износа у инструментов с системами резцов)
 - SD42930 \$SC_WEAR_SIGN (знак износа)
 - SD42935 \$SC_WEAR_TRANSFORM (трансформации для компонентов инструмента)
 - SD42940 \$SC_LENGTH_CONST (смена компонентов длин инструмента при смене плоскости)
 - SD42942 \$SC_TOOL_LENGTH_CONST_T (смена компонентов длин инструмента для токарных инструментов при смене плоскости)
 - SD42950 \$SC_TOOL_LENGTH_TYPE (соотнесение компонентов длин инструмента независимо от типа инструмента)
 - SD42954 \$SC_TOOL_ORI_CONST_M (смена компонентов ориентации инструмента для фрезерных инструментов при смене плоскости)
 - SD42956 \$SC_TOOL_ORI_CONST_T (смена компонентов ориентации инструмента для токарных инструментов при смене плоскости)
- Компонент ориентации текущего общего фрейма (вращения и отражения, не смещения нулевой точки или отражения)
- Компонент ориентации и результирующая длина активного ориентируемого инструментального суппорта
- Компонент ориентации и результирующая длина активной трансформации

Кроме названных данных, описывающих окружение инструмента, сохраняются и номера T, номера D и номера DL активного инструмента, чтобы после можно было обратиться к этому инструменту в том же окружении, что и при вызове TOOLENV без повторного обозначения инструмента.

Синтаксис

<состояние> = TOOLENV (<имя>)

Объяснение

TOOLENV (. . .) :	Предопределенная функция сохранения инструментального окружения	
	Автономно в кадре:	да

<состояние>:		Возвращаемое значение функции. Отрицательные значения показывают состояния ошибки.		
		Тип данных: INT		
		Значение:	0	Функция ОК
			-1	не зарезервировано места в памяти для инструментального окружения: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Т. е. функциональность "Инструментальные окружения" отсутствует.
			-2	более нет свободного места для инструментальных окружений в памяти.
			-3	нулевая строка как имя инструментального окружения не допускается.
			-4	Параметр (<имя>) не указан.
Параметры				
1	<имя>:	Имя, под которым необходимо сохранять текущий блок данных. Если блок данных с тем же именем уже имеется, то он заменяется. В этом случае состояние 0.		
		Тип данных:	STRING	

Дополнительная информация

Базовый размер / размер адаптера — коррекция на длину инструмента

Входит ли длина адаптера или базовый размер (спец. для резца параметры \$TC_DP21, \$TC_DP22, \$TC_DP23) в вычисление длины инструмента, определяется при активном управлении магазином инструментов (доступно только с опцией "Магазин инструментов"!) по значению в машинных данных:

MD18104 \$MN_MM_NUM_TOOL_ADAPTER (адаптер инструмента в области TO)

Поскольку изменение этих машинных данных может начать действовать только запуске СЧПУ, оно в инструментальном окружении не сохраняется.

Результирующая длина ориентируемых инструментальных суппортов и трансформаций

Примечание

Как при ориентируемых инструментальных суппортах, так и при трансформациях, существуют системные переменные или машинные данные, которые действуют как дополнительные компоненты длин инструмента, и которые полностью или частично подвержены этим исполняемым вращениям. Дополнительные компоненты длин инструмента, получаемые вследствие этого, должны также быть сохранены при вызове TOOLENV, т. к. они являются частью окружения, в котором используется инструмент.

Трансформация адаптера

Трансформация адаптера это свойство адаптера инструмента и тем самым всего инструмента. Поэтому она не является составной частью инструментального окружения, которая может быть применена к другому инструменту.

Посредством сохранения всех необходимых для определения общей длины инструмента данных позднее можно вычислить эффективную длину инструмента, даже если на этот момент он более не активен или если условия окружения (например, G-команды или установочные данные) изменились. Также можно вычислить эффективную длину другого инструмента с допущением, что он использовался в тех же условиях, что и инструмент, для которого было сохранено состояние.

Макс. число наборов данных для инструментальных окружений

Машинный параметр MD18116 \$MN_MM_NUM_TOOL_ENV определяет, какое максимальное количество блоков данных можно сохранить для описания инструментальных окружений. Данные находятся в диапазоне TOA. Они сохраняются и после отключения СЧПУ.

Сохранение данных невозможно. Это означает, что эти данные невозможно передать на другую систему управления.

11.12.2 Удаление инструментального окружения (DELTOOLENV)

С помощью функции DELTOOLENV блоки данных описания инструментальных окружений можно удалить. Удаление означает, что обращение к сохраненному под определенным именем блоку данных более невозможно (попытка обращения приводит к аварийному сообщению).

Примечание

Блоки данных можно удалять только функцией DELTOOLENV, через загрузку INITIAL.INI или через холодный пуск (запуск ЧПУ со стандартными машинными данными). Других автоматических процессов удаления нет.

Синтаксис

<состояние> = DELTOOLENV (<имя>)
<состояние> = DELTOOLENV ()

Объяснение

DELTOOLENV (. . .) :	Предопределенная функция удаления инструментального окружения	
	Автономно в кадре:	да

<состояние>:		Возвращаемое значение функции. Отрицательные значения показывают состояния ошибки.		
		Тип данных:	INT	
		Значение:	0	Функция ОК
			-1	не зарезервировано места в памяти для инструментального окружения: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Т. е. функциональность "Инструментальные окружения" отсутствует.
		-2	инструментальное окружение с указанными именем не существует.	
Параметры				
1	<имя>:	Имя удаляемого блока данных		
		Тип данных:	STRING	
DELTOOLENV () :		DELTOOLENV () без указания имени удаляет все блоки данных описания инструментального окружения		

11.12.3 Чтение номеров T, D и DL (GETTENV)

Функция GETTENV служит для чтения сохраненных в инструментальном окружении номеров T, D и DL.

Синтаксис

<состояние> = GETTENV (<Name>, <TDDL>)

Объяснение

GETTENV (. . .) :		Предопределенная функция чтения номеров T, D, DL в блоке данных описания инструментального окружения	
		Автономно в кадре:	да
<состояние>:		Возвращаемое значение функции. Отрицательные значения показывают состояния ошибки.	
		Тип данных:	INT
		Значение:	0 Функция ОК
			-1 не зарезервировано места в памяти для инструментального окружения: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Т. е. функциональность "Инструментальные окружения" отсутствует.
			-2 инструментальное окружение с указанными именем не существует.
Параметры			

1	<имя>:	Имя блока данных, из которого необходимо считывать номера T, D, DL.	
		Тип данных:	STRING
2	<TDDL>:	В поле этого параметра результата содержатся номера T, D, DL инструмента, инструментальное окружение которого сохранено в указанном блоке данных: • <TDDL> [0]: номер T • <TDDL> [1]: номер D • <TDDL> [2]: номер DL	
		Тип данных:	INT[3]
GETTENV (, <TDDL>) , GETTENV (" " , <TDDL>) :		Допускается пропустить при вызове функции GETTENV первый параметр или передать в качестве первого параметра нулевую строку. В обоих этих специальных случаях в <TDDL> возвращаются номера T, D, DL активного инструмента.	

11.12.4 Считывание данных сохраняемого инструментального окружения (\$P_TOOLENVN, (\$P_TOOLENV)

Данные сохраняемого инструментального окружения считываются по следующим системным переменным:

\$P_TOOLENVN:	Выводит количество определенных с помощью TOOLENV (и еще не удаленных) блоков данных для описания инструментальных окружений.			
	Синтаксис:	<n> = \$P_TOOLENVN		
	Объяснение:	<n>:	Количество определенных блоков данных	
			Тип данных:	INT
			Диапазон значений:	0 ... MD18116 \$MN_MM_NUM_TOOL_ENV
Обращение к этой системной переменной возможен и тогда, когда инструментальные окружения невозможны (MD18116 = 0). В этом случае возвращаемое значение — "0".				

\$P_TOOLENV:	Выводит имя i-го блока данных описания инструментального окружения.			
	Синтаксис:	<Name> = \$P_TOOLENV[<i>]		
	Объяснение:	<имя>:	Имя блока данных с номером <i>.	
			Тип данных:	STRING
		<i>:	Номер блока данных	
			Тип данных:	INT
			Диапазон значений:	1 ... \$P_TOOLENVN
			Согласование номеров с блоками данных не является фиксированным, а может изменяться через удаление и повторное создание блоков данных. Нумерация блоков данных является внутренней.	
	Если <i> ссылается на не определенный блок данных, возвращается нулевая строка.			
	Если индекс <i> недействителен, т. е. <i> меньше 1 или больше значения максимального количества блоков данных для инструментальных окружений (MD18116 \$MN_MM_NUM_TOOLENV), выводится следующее аварийное сообщение:			
Аварийное сообщение "17020 (неразрешенный индекс массива1)"				

11.12.5 Считывание компонентов длины инструмента или компонентов длины (GETTCOR)

Функция GETTCOR служит для считывания длин инструмента или компонентов длин инструмента.

При этом посредством параметрирования можно указать, какие компоненты необходимо учесть, и при каких условиях использования необходимо рассматривать инструмент.

Синтаксис

<состояние> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])

Объяснение

GETTCOR(...):	Предопределенная функция чтения длин инструмента или компонентов длин инструмента	
	Автономно в кадре:	да

<состояние>:	Возвращаемое значение функции. Отрицательные значения показывают состояния ошибки.		
	Тип данных:	INT	
	Значе- ние:	0	Функция ОК
		-1	не зарезервировано места в памяти для инструментально- го окружения: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Т. е. функциональность "Инструментальные окружения" от- сутствует.
		-2	Инструментальное окружение с указанным в <Stat> именем не существует.
		-3	Недействительная строка в параметре <Comp>. Причиной этой ошибки могут быть недействительные или запрограммированные дважды символы.
		-4	недействительный номер T
		-5	недействительный номер D
		-6	недействительный номер DL
		-7	Попытка обращения к отсутствующему модулю памяти
		-8	Попытка обращения к отсутствующей опции (например, программируемая ориентация инструмента, управление инструментом)
		-9	Строка <Comp> содержит двоеточие (идентификатор спе- цификаций системы координат), но за ним не следует дей- ствительный символ обозначения требуемой системы коор- динат.
Параметры			

1	<Len>:	Результирующий вектор	
		Тип данных:	REAL[11]
Компоненты вектора расположены в следующей последовательности:			
• <Len> [0]: Тип инструмента • <Len> [1]: Положение резцов • <Len> [2]: Абсцисса • <Len> [3]: Ордината • <Len> [4]: Аппликата • <Len> [5]: Радиус инструмента			
Исходной системой координат для компонентов длин является определенная в <Comp> и <Stat> система координат. Если в <Comp> система координат не определена, то длины инструмента отображаются в системе координат станка.			
Соотнесение абсциссы, ординаты и аппликаты с гео-осями зависит от активной плоскости в используемом инструментальном окружении. Т. е. при G17 абсцисса параллельна X, при G18 — Z и т. д.			
Компоненты с <Len>[6] по <Len>[10] содержат дополнительные параметры, которые могут быть указаны для описания геометрии инструмента (например, с \$TC_DP7 по \$TC_DP11 для геометрии или соответствующие компоненты для износа или суммарной и отладочной коррекции).			
Эти 5 дополнительных элементов и радиус инструмента определены только для компонентов E, G, S и W. Их нормирование не зависит от <Stat>. Соответствующие значения в компонентах с <Len>[6] по <Len>[10] могут отличаться от нуля, только если в вычислении длин инструмента участвует, по меньшей мере, один из четырех названных компонентов. Прочие компоненты на результат не влияют. Данные размеров относятся к основной системе СЧПУ (дюймовая или метрическая).			

2	<Comp>:	Компоненты длин инструмента (опционально)			
		Тип данных:	STRING		
		Эта цепочка символов состоит из двух частей, разделенных двоеточием.			
		Общая форма: "<SubStr_1> [: <SubStr_2>]"			
		<SubStr_1>:	Первая часть строки обозначает компоненты длин инструмента, которые необходимо учитывать при вычислении длин инструмента.		
			Последовательность символов в части строки и их написание (прописное или строчное) — произвольные. Между символами можно вставлять любое число пробелов или символов табулятора (white spaces).		
			Указание: Двойное программирование символов в части строки не допускается.		
			Символ:	-	знак минус (допустим только как первый символ!) Вычисляется вся длина инструмента, за исключением компонентов, указанных в следующей строке.
				C	размер адаптера или базовый размер (один из двух существующих как альтернатива компонентов, который активен для используемого инструмента)
				E	отладочные коррекции
G	геометрия				
K	Кинематическая трансформация (обрабатывается только при базовой 3-, 4- и 5-осевой трансформации)				
S	Суммарные коррекции				
T	Ориентируемый инструментальный суппорт				
W	Износ				
Если первая часть строки (за исключением white spaces) пустая, то это означает, что всю длину инструмента необходимо вычислять с учетом всех компонентов. То же действительно, и если параметр <Comp> не указывается.					
<Substr_2>:	Оptionальная вторая часть строки обозначает систему координат, в которой должна выводиться длина инструмента.				
	Вторая часть строки состоит только из одного релевантного символа.				
	Символ:	A	Настраиваемая система координат (ACS)		
		B	Базовая кинематическая система (BKS)		
		K	Система координат инструмента кинематической трансформации (KCS)		
		M	Система координат станка (MCS)		
		T	Система координат инструмента (TCS)		
W		Система координат детали (WCS)			

		Если система координат не указывается, то нормирование выполняется в системе координат станка (MCS). Учитываемые при необходимости вращения устанавливаются через определенное в <Stat> инструментальное окружение.
3	<_Stat>:	Имя блока данных для описания инструментального окружения (опционально) Тип данных: STRING Если значение этого параметра нулевая строка ("") или оно не указывается, то используется текущее состояние. Если инструмент не задан, используется текущий инструмент.
4	<T>:	Внутренний номер T инструмента (опционально) Тип данных: INT Если этот параметр не указан или если его значение 0, то используется сохраненный в <Stat> инструмент. Если значение этого параметра "-1", то используется номер T активного инструмента. Допускается явное указание номера активного инструмента. Указание: Если <Stat> не указана, то в качестве инструментального окружения используется текущее состояние. Поскольку с <T> = 0 выполняется ссылка на сохраненный в инструментальном окружении номер T, в нем используется активный инструмент, т. е. данные <T> = 0 и <T> = -1 в этом особом случае равнозначны.
5	<D>:	Номер резца инструмента (опционально) Тип данных: INT Если этот параметр не указан или его значение 0, то используемый номер D зависит от источника номера T. Если используется номер T из инструментального окружения, то считывается и номер D инструментального окружения, в ином случае таковой текущего активного инструмента.
6	<DL>:	Номер зависящей от места коррекции (опционально) Тип данных: INT Если этот параметр не указан, то используемый номер DL зависит от источника номера T. Если используется номер T из инструментального окружения, то считывается и номер D инструментального окружения, в ином случае таковой текущего активного инструмента.

Примеры

GETTCOR (_LEN)	Вычисляет длину текущего активного инструмента в системе координат станка с учетом всех компонентов.
GETTCOR (_LEN , "CGW:W")	Вычисляется длина инструмента, состоящая из размера адаптера или базового размера, геометрии и износ для активного инструмента. Прочие компоненты, например, ориентированный инструментальный суппорт или кинематические трансформации, не учитываются. Вывод выполняется в системе координат детали.

GETTCOR (_LEN, "-K:B")	Вычисляется общая длина активного инструмента без учета компонентов длин возможно активной кинематической трансформации. Вывод в базовой кинематической системе.
GETTCOR (_LEN, ":M", "Testenv1", , 3)	Вычисляется общая длина инструмента для сохраненного в инструментальном окружении под именем "Testenv1" инструмента в системе координат станка. Но вычисление выполняется независимо от сохраненного номера резца для номера резца D3.

Дополнительная информация

Трансформация адаптера / ориентируемые инструментальные суппорты / кинематическая трансформация

Вращения и смены компонентов, которые в некоторых случаях выполняются трансформацией адаптера, ориентируемым инструментальным суппортом и кинематической трансформацией, являются составной частью инструментального окружения. Поэтому они выполняются всегда, и тогда, когда соответствующий компонент длины не должен учитываться. Если это нежелательно, то необходимо определить инструментальные окружения, в которых соответствующие трансформации не активны. Во многих случаях (а именно всегда, когда на станке не используются трансформации или ориентируемые инструментальные суппорты), сохраненные блоки данных для инструментального окружения выполняют эти условия автоматически, поэтому отдельного рассмотрения их пользователем не требуется.

Токарные и шлифовальные инструменты: Вычисление длины инструмента в зависимости от MD20360 \$MC_TOOL_PARAMETER_DEF_MASK

В машинных данных определяется, как у токарного и шлифовального инструмента следует нормировать износ или длину инструмента в, возможно, имеющейся оси диаметра.

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (определение параметров инструмента)

Бит	Значение	
0	В случае токарного и шлифовального инструмента параметр износа поперечной оси учитывается при вычислении как значение диаметра:	
	= 0 (по умолчанию)	нет
	=1	да
1	В случае токарного и шлифовального инструмента компонент длины инструмента поперечной оси учитывается при вычислении как значение диаметра:	
	= 0 (по умолчанию)	нет
	=1	да

Если соответствующие биты установлены, то соответствующая строка нормируется с коэффициентом 0.5. Такое нормирование отражается и в возвращенной GETTCOR длине инструмента.

Пример:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 \$MC_DIAMETER_AX_DEF (гео-ось с функцией поперечной оси) = "X"

X — это ось диаметра (стандартная конфигурация токарных станков)

Программный код	Комментарий
N30 \$TC_DP1[1,1]=500	
N40 \$TC_DP2[1,1]=2	
N50 \$TC_DP3[1,1]=3.0	; геометрия L1
N60 \$TC_DP4[1,1]=4.0	
N70 \$TC_DP5[1,1]=5.0	
N80 \$TC_DP12[1,1]=12.0	; износ L1
N90 \$TC_DP13[1,1]=13.0	
N100 \$TC_DP14[1,1]=14.0	
N110 T1 D1 G18	
N120 R1=GETTCOR(_LEN, "GW")	
N130 R3=_LEN[2]	; 17.0 (= 4.0 + 13.0)
N140 R4=_LEN[3]	; 7.5 (= 0.5 * 3.0 + 0.5 * 12.0)
N150 R5=_LEN[4]	; 19.0 (= 5.0 + 14.0)
N160 M30	

Компоненты длины кинематической трансформации и ориентируемого инструментального суппорта

Если при вычислении длин инструмента учитывается **ориентируемый инструментальный суппорт**, то следующие векторы включаются в вычисление длин инструмента:

Тип	Векторы
M	I1 и I2
T	I1, I2 и I3
P	Ориентируемый инструментальный суппорт не влияет на длину инструмента.

При базовой **5-осевой трансформации** при типах трансформации 24 и 56 следующие машинные данные включаются в вычисление длин инструмента:

Тип трансформации	Машинные данные
24	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 MD24558/24658 \$MC_TRAFO5_PART_OFFSET_1/2
56	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2

Тип трансформации 56 (подвижный инструмент и подвижная деталь) соответствует типу M для ориентируемого инструментального суппорта.

При этой 5-осевой трансформации в прежних версиях ПО вектор MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 (вектор кинематического смещения 1-ой/2-ой 5-осевой трансформации в канале) соответствует сумме обоих векторов I₁ и I₃ у ориентируемого инструментального суппорта типа M:

В обоих случаях для трансформации важна только сумма. Тип структуры из двух отдельных компонентов не имеет значения. Но при вычислении длины инструмента имеет значение, какой компонент согласован с инструментом, а какой – с инструментальным столом. Поэтому были введены машинные данные MD24558/24658 \$MC_TRAFO5_JOINT_OFFSET_PART_1/2 (вектор кинематического смещения в столе). Он соответствует вектору I3. Теперь машинные данные MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 больше не соответствуют сумме I1 и I3, а соответствуют только вектору I1. Если машинные данные MD24558/24658 \$MC_TRAFO5_JOINT_OFFSET_PART_1/2 равны нулю, то поведение идентично прежнему.

Совместимость

Функция GETTCOR в комбинации с функциями TOOLENV и SETTCOR служит, среди прочего, для замены частей функциональности, прежде реализованных отдельно в измерительных циклах.

В измерительных циклах обрабатывалась только часть параметров, которые в конечном итоге определяют эффективную длину инструмента. Названные функции можно спараметризовать так, что появляется возможность воспроизвести поведение измерительных циклов в отношении вычисления длин инструмента.

11.12.6 Изменение компонентов инструмента (SETTCOR)

Функция SETTCOR изменяет компоненты инструмента с учетом всех граничных условий, входящих в нормирование отдельных компонентов.

Примечание

Терминология: Если далее в связи с длиной инструмента речь идет о компонентах инструмента, то подразумеваются компоненты, рассматриваемые в векторной форме, из которых состоит полная длина инструмента (например, геометрия или износ). Такой компонент состоит из трех отдельных значений (L1, L2, L3), обозначаемых далее "значения координат".

Компонент инструмента "геометрия" состоит, например, из трех значений координат от \$TC_DP3 до \$TC_DP5.

Синтаксис

```
<состояние> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>, <GeoAx>, <Stat>, <T>, <D>, <DL>])
```

Объяснение

SETTCOR(...):	Предопределенная функция изменения компонентов инструмента	
	Автономно в кадре:	да

<состояние>:		Возвращаемое значение функции. Отрицательные значения показывают состояния ошибки.		
		Тип данных: INT		
		Значение:	0	Функция ОК
			-1	Не зарезервировано места в памяти для инструментального окружения: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 Т. е. функциональность "Инструментальные окружения" отсутствует.
			-2	Инструментальное окружение с указанным в <Stat> именем не существует.
			-3	Недействительная строка в параметре <Comp>. Причиной этой ошибки могут быть недействительные или запрограммированные дважды символы.
			-4	Недействительный номер T
			-5	Недействительный номер D
			-6	Недействительный номер DL
			-7	Попытка обращения к отсутствующему модулю памяти
			-8	Попытка обращения к отсутствующей опции (например, программируемая ориентация инструмента, управление инструментом)
			-9	Недопустимое числовое значение параметра <CorComp>.
			-10	Недопустимое числовое значение параметра <CorMode>.
			-11	Содержание параметров <Comp> и <CorComp> противоречиво.
			-12	Содержание параметров <Comp> и <CorMode> противоречиво.
-13	Содержание параметра <GeoAx> не обозначает гео-ось.			
-14	Попытка записи на отсутствующую отладочную коррекцию			
Параметры				
1	<CorVal>:	Вектор коррекции В определенной в <Stat> системе координат детали (WCS) действует следующее соотношение: <CorVal> [0]: Абсцисса <CorVal> [1]: Ордината <CorVal> [2]: Аппликата Если необходимо исправить только один компонент инструмента (например, невекторная коррекция, см. параметр <CorMode>), то значение коррекции всегда находится в <CorVal>[0] независимо от того, для какой оси он действует. При этом содержание двух остальных компонентов не обрабатывается. Если <CorVal> или компонент <CorVal> относится к поперечной оси, то данные нормируются как размер радиуса. Это означает, что инструмент удлинится, например, на заданный размер, и это ведет, соответственно, к двойному увеличению диаметра детали. Данные размеров относятся к основной системе СЧПУ (дюймовая или метрическая).		
		Тип данных:	REAL[3]	

2	<Comp>:	Компонент(ы) инструмента					
		Тип данных:	STRING				
		Эта цепочка символов состоит из двух частей, разделенных двоеточием.					
		Общая форма: "<SubStr_1> [: <SubStr_2>]"					
		<SubStr_1>:	Первая часть строки всегда должна быть в наличии и может состоять из одного или из двух символов. При этом первый или единственный символ обозначает 1-й компонент (Val ₁), а второй символ — 2-й компонент (Val ₂), обрабатываемые в соответствии со следующими параметрами <CorComp> и <CorMode>.				
					Символ:	C	размер адаптера или базовый размер (один из двух существующих как альтернатива компонентов, который активен для используемого инструмента)
						E	отладочные коррекции
						G	геометрия
						S	суммарные коррекции
						W	износ
<Substr_2>:	Вторая часть строки опциональна. Она может состоять на выбор из (единственной) буквы "W" или "T".						
			Символ:	W	Если вторая часть строки пустая или содержит букву "W", то значения коррекции вычисляются так, как если бы их измеряли в системе координат детали (WCS).		
				T	Если вторая часть строки содержит букву "T", то значения коррекции вычисляются так, как если бы их измеряли в системе координат инструмента (Tool Coordinate System, TCS).		
Написание символов в строке (прописное или строчное) является произвольным. Можно вставлять любое число пробелов или символов табулятора (white spaces).							

3	<CorComp>:	Задает компонент(ы) блока данных инструмента, которые необходимо описать (опционально)		
		Тип данных:	INT	
		Значение:	0	Значение коррекции <CorVal>[0] относится к гео-оси, определенной в параметре <GeoAx>, в системе координат детали или в системе координат инструмента (см. описание параметра <Comp>). Это означает, что значение коррекции должно быть так учтено в обозначенных компонентах инструмента, что с учетом всех параметров, способных повлиять на вычисление длин инструмента, результатом оказывается изменение всей длины инструмента в указанном осевом направлении на заданное значение. Такое изменение можно получить коррекцией указанных в <Comp> компонентов и указанного в <CorMode> правила вычисления символов (см. следующий параметр). Поэтому получившаяся коррекция может сказываться на всех трех компонентах оси.
			1	Как "0", но векторн. Содержание вектора <CorVal> относится к абсциссе, ординате и аппликате в системе координат детали или в системе координат инструмента (см. описание параметра <Comp>). Следующий параметр <GeoAx> не обрабатывается.
			2	Векторная коррекция, т.е., L1, L2, L3 можно изменять одновременно. Однако в отличие от вариантов "0" и "1" содержащиеся в <CorVal> значения коррекции относятся к координатам компонента Val _i (см. следующий параметр <CorMode>) инструмента. Возможное наклонное положение инструмента относительно системы координат детали на коррекцию не влияет.
			3 – 5	Коррекция значений длины инструмента с L1 по L3 (с \$TC_DP3 по \$TC_DP5) или соответствующих значений для коррекции на износ, отладочной коррекции или суммарной коррекции. Значение коррекции содержится в <CorVal>[0]. Оно измерено в координатах компонента Val _i (см. следующий параметр <CorMode>). Возможное наклонное положение инструмента относительно системы координат детали на коррекцию не влияет.
			6	Коррекция значения радиуса инструмента (\$TC_DP6) или соответствующих значений для коррекции на износ, отладочной коррекции или суммарной коррекции. Учитываются биты 10 и 11 (нормирование указания диаметра или износа диаметра на выбор как указание радиуса или диаметра) в машинных данных MD20360 \$MC_TOOL_PARAMETER_DEF_MASK.
			7 – 11	Коррекция с \$TC_DP7 по \$TC_DP11 или соответствующих значений для коррекции на износ, отладочной коррекции или суммарной коррекции. Эти параметры обрабатываются как радиус инструмента.
Если этот параметр не указан, его значение — "0".				

4

<CorMode>:

Задаёт тип выполняемой операции записи (опционально)		
Тип данных:	INT	
Значение:	0	$Val_{1neu} = <CorVal>$
	1	$Val_{1neu} = Val_{1alt} + <CorVal>$
	2	$Val_{1neu} = <CorVal>$ $Val_{2neu} = 0$
	3	$Val_{1neu} = Val_{1alt} + Val_{2alt} + <CorVal>$ $Val_{2neu} = 0$

Форму записи $Val_{1alt} + Val_{2alt}$ следует понимать как символическую. Если оба компонента (на основании состояния $<_Stat>$) нормированы различно, т. е. если между обоими компонентами действует вращение, то перед сложением Val_{2alt} преобразуется таким образом, что получающаяся длина инструмента после удаления Val_{2neu} и перед сложением $<CorVal>$ остается неизменной.

$<CorVal>$ всегда относится к Val_1 . $<CorVal>$ — это значение, измеряемое в зависимости от второй части параметра $<Comp>$ в системе координат детали (WCS) или в системе координат инструмента (TCS). Поэтому по отношению к компоненту инструмента, для которого его необходимо учесть, оно в соответствующих случаях уже преобразовано. Поэтому его невозможно вычислить с сохраненным значением: перед сложением необходимо предпринять обратное преобразование в значение Val_1 или Val_2 . Это может привести к тому, что коррекция сказывается на другой оси, а не оси, определенной посредством $<CorComp>$, или на нескольких осях.

Если $<CorComp> = 0$, т. е., если $<CorVal>$ содержит не вектор, а лишь отдельное значение, описанные операции выполняются в координатах замера $<CorVal>$ (WCS / TCS). В частности, это действительно для установления на нуль Val_{2neu} в вариантах 2 и 3. Затем этот результат преобразуется обратно в координаты инструмента. Это может привести к тому, что ни одно из установленных на нуль значений координат (L_1 , L_2 , L_3) не равно нулю, или к тому, что значения координат, которые ранее были равны нулю, теперь отличны от нуля. Однако если соответствующие операции выполняются последовательно для всех трех гео-осей, есть гарантия того, что все три значения координат удаляемого компонента будут равны нулю. Если инструмент не повернут по отношению к системе координат детали или повернут так, что все компоненты инструмента остаются параллельными к осям координат (перестановки осей), есть гарантия того, что изменяется соответственно только одна координата инструмента.

Последовательное выполнение одинаковой операции ($<CorMode>$) при $<CorComp> = 0$ для всех трех осей координат в произвольной последовательности идентично однократному выполнению одинаковой операции при $<CorComp>=1$.

Для значений параметров "0" и "1" параметр $<Comp>$ должен содержать один символ, для значений параметров "2" и "3" — два символа.

Пример:

$<Comp>$ содержит строку "ES", $<CorMode>$ содержит значение "2"

$\Rightarrow$ отладочная коррекция_{нов} = $<CorVal>$, суммарная коррекция_{нов} = 0

Если параметр $<CorMode>$ не указан, его значение — "0".

5	<GeoAx>:	Указывает индекс гео-оси, в который измерялось значение коррекции <CorVal>[0] (опционально)	
		Тип данных:	INT
		Диапазон значений:	0 ... 2
		Индексы от 0 до 2 относятся к абсциссе, ординате и аппликате в действующей плоскости (G17/G18/G19) текущего инструментального окружения. Содержание этого параметра обрабатывается, только если параметр <CorComp> имеет значение "0".	
6	<Stat>:	Имя блока данных для описания инструментального окружения (опционально)	
		Тип данных:	STRING
		Если значение этого параметра нулевая строка (""), или оно не указывается, то используется текущее состояние. Если инструмент не задан, используется текущий инструмент.	
7	<T>:	Внутренний номер T инструмента (опционально)	
		Тип данных:	INT
		Если этот параметр не указан, или если его значение 0, то используется сохраненный в <Stat> инструмент.	
		Если значение этого параметра "-1", то используется номер T активного инструмента. Допускается явное указание номера активного инструмента.	
8	<D>:	Указание: Если <Stat> не указана, то в качестве инструментального окружения используется текущее состояние. Поскольку с <T> = 0 выполняется ссылка на сохраненный в инструментальном окружении номер T, в нем используется активный инструмент, т. е. данные <T> = 0 и <T> = -1 в этом особом случае равнозначны.	
		Номер резца инструмента (опционально)	
		Тип данных:	INT
		Если этот параметр не указан или его значение 0, то используемый номер D зависит от источника номера T. Если используется номер T из инструментального окружения, то считывается и номер D инструментального окружения, в ином случае таковой текущего активного инструмента.	
9	<TL>:	Номер зависящей от места коррекции (опционально)	
		Тип данных:	INT
		Если этот параметр не указан, то используемый номер DL зависит от источника номера T. Если используется номер T из инструментального окружения, то считывается и номер D инструментального окружения, в ином случае таковой текущего активного инструмента. Если T, D, DL задают инструмент без зависящих от местоположения коррекций, то в параметре <Comp> не должны задаваться суммарные или отладочные коррекции (код ошибки в <Status>).	

Примечание

Не все возможные сочетания трех параметров <Comp>, <CorComp> и <CorMode> имеют смысл. Например, правило вычисления 3 требует в <CorComp> указания двух знаков в <Comp>. Если указывается недопустимое сочетание параметров, в <Status> возвращается соответствующий код ошибки.

Примеры

Пример 1

Программный код	Комментарий
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; фрезерный инструмент
N30 \$TC_DP3[1,1]=10.0	; геометрия L1
N40 \$TC_DP12[1,1]=1.0	; износ L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"G",0,0,2)	
N80 T1 D1 X0 Y0 Z0	; ==> позиция MCS X0.000 Y0.000 Z1.333
N90 M30	

<CorComp> = "0", поэтому действующее в направлении Z значение координаты геометрического компонента заменяется на значение коррекции 0,333.

Таким образом, результирующая общая длина инструмента: $L1 = 0,333 + 1,000 = 1,333$

Пример 2

Программный код	Комментарий
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; фрезерный инструмент
N30 \$TC_DP3[1,1]=10.0	; геометрия L1
N40 \$TC_DP12[1,1]=1.0	; износ L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"W",0,1,2)	
N80 T1 D1 X0 Y0 Z0	; ==> позиция MCS X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> = "1", поэтому действующее в направлении Z значение коррекции 0,333 прибавляется к значению износа 1,0.

Таким образом, результирующая общая длина инструмента: $L1 = 10,0 + 1,333 = 11,333$

Пример 3

Программный код	Комментарий
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; фрезерный инструмент
N30 \$TC_DP3[1,1]=10.0	; геометрия L1
N40 \$TC_DP12[1,1]=1.0	; износ L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,2,2)	
N80 T1 D1 X0 Y0 Z0	; ==> позиция MCS X0.000 Y0.000 Z0.333
N90 M30	

<CorComp> = 2, поэтому действующая в направлении Z коррекция вносится в геометрический компонент (старое значение заменяется), и значение износа удаляется.

Таким образом, результирующая общая длина инструмента: $L1 = 0,333 + 0,0 = 0,333$

Пример 4

Программный код	Комментарий
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; фрезерный инструмент
N30 \$TC_DP3[1,1]=10.0	; геометрия L1
N40 \$TC_DP12[1,1]=1.0	; износ L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,3,2)	
N80 T1 D1 X0 Y0 Z0	; ==> позиция MCS X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> = 3, поэтому значение износа и значение коррекции прибавляется к геометрическому компоненту, и компонент износа удаляется.

Таким образом, результирующая общая длина инструмента: $L1 = 11,333 + 0,0 = 11,333$.

Пример 5

Программный код	Комментарий
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; фрезерный инструмент
N30 \$TC_DP3[1,1]=10.0	; геометрия L1
N40 \$TC_DP12[1,1]=1.0	; износ L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,3,0)	
N80 T1 D1 X0 Y0 Z0	; ==> позиция MCS X0.333 Y0.000 Z11.000
N90 M30	

<CorComp> = 3, как в предыдущем примере, но теперь коррекция действует на гео-ось с индексом "0" (ось X), с которой из-за G17 у фрезерного инструмента соотнесен компонент инструмента L3. Поэтому вызов SETTCOR не влияет на параметры инструмента \$TC_DP3 и \$TC_DP12. Вместо этого значение коррекции вносится в \$TC_DP5.

Пример 6

Программный код	Комментарий
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; токарный инструмент
N30 \$TC_DP3[1,1]=10.0	; геометрия L1
N40 \$TC_DP4[1,1]=15.0	; геометрия L2
N50 \$TC_DP12[1,1]=10.0	; износ L1
N60 \$TC_DP13[1,1]=0.0	; износ L2

Программный код	Комментарий
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",0,3,1)	
N110 T1 D1 X0 Y0 Z0	; ==> позиция MCS X24.330 Y0.000 Z17.500
N120 M30	

Инструмент — это токарный инструмент. В кадре N80 активируется вращение фрейма, поэтому базовая кинематическая система (BCS) повернута по отношению к системе координат детали (WCS). Значение коррекции (N70) воздействует в WCS на гео-ось с индексом "1", т. е., поскольку G18 активна, на ось X. Поскольку <CorMode> = 3, износ инструмента в направлении оси X WCS после выполнения N100 должен стать равным нулю.

Поэтому содержание релевантных параметров инструмента в конце программы:

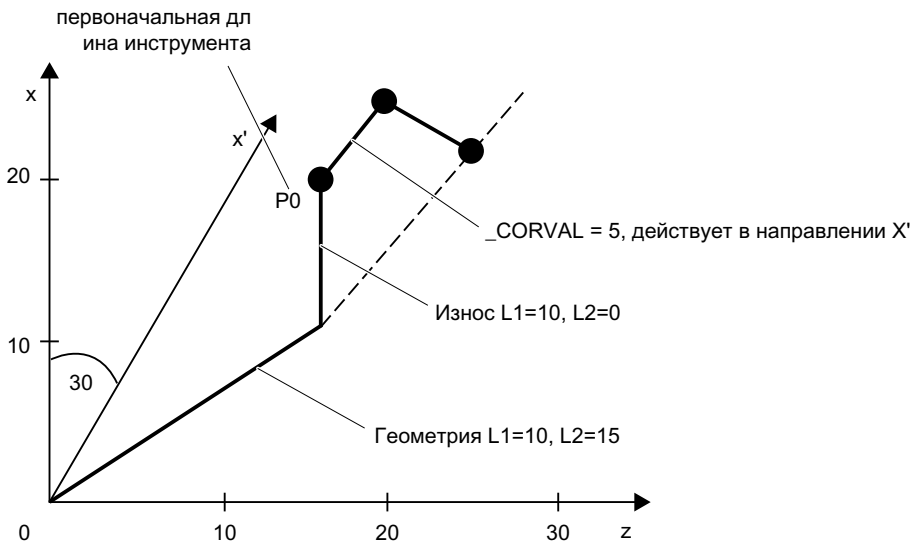
\$TC_DP3[1,1] : 21.830 ; геометрия L1

\$TC_DP4[1,1] : 21.830 ; геометрия L2

\$TC_DP12[1,1] : 2.500 ; износ L1

\$TC_DP13[1,1] : -4.330 ; износ L2

Геометрические соотношения представлены на следующем изображении. Весь износ, включая _CORVAL, проецируется на направление X' в WCS. Это дает точку P2. Координаты этой точки (измерены в координатах X-Y) вносятся в геометрический компонент инструмента. В износе остается дифф. вектор $P_2 - P_1$. Тем самым у износа больше нет компонента в направлении _CORVAL.



Если иллюстративная программа продолжается после N110 со следующими операторами, то оставшийся износ полностью передается в геометрию, поэтому коррекция теперь действует в оси Z' (параметр <GeoAx> = 0).

N120 _CORVAL[0]=0.0

```
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
```

```
N140 T1 D1 X0 Y0 Z0 ; ==> позиция MCS X24.330 Y0.000 Z17.500
```

Поскольку новое значение коррекции равно 0, изменение общей длины инструмента и тем самым позиции подвода в N140 запрещено. Если бы _CORVAL в N120 была отлична от 0, то получилась бы новая общая длина инструмента, и тем самым и измененная позиция в N140, но компонент износа длины инструмента был бы в любом случае равен нулю, т. е. общая длина инструмента в результате в любом случае содержится в геометрическом компоненте инструмента.

Результат, идентичный двукратному вызову функции SETTCOR с параметром <CorComp> = 0, достигается и посредством однократного вызова с <CorComp> = 1 (векторная коррекция):

Программный код	Комментарий
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; токарный инструмент
N30 \$TC_DP3[1,1]=10.0	; геометрия L1
N40 \$TC_DP4[1,1]=15.0	; геометрия L2
N50 \$TC_DP12[1,1]=10.0	; износ L1
N60 \$TC_DP13[1,1]=0.0	; износ L2
N70 _CORVAL[0]=0.0	
N71 _CORVAL[1]=5.0	
N72 _CORVAL[2]=0.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",1,3,1)	
N110 T1 D1 X0 Y0 Z0	; ==> позиция MCS X24.330 Y0.000 Z17.500
N120 M30	

В этом случае все компоненты износа инструмента сразу же после первого вызова SETTCOR в N100 равны нулю.

Пример 7

Программный код	Комментарий
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; токарный инструмент
N30 \$TC_DP3[1,1]=10.0	; геометрия L1
N40 \$TC_DP4[1,1]=15.0	; геометрия L2
N50 \$TC_DP12[1,1]=10.0	; износ L1
N60 \$TC_DP13[1,1]=0.0	; износ L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",3,3)	
N110 T1 D1 X0 Y0 Z0	; ==> позиция MCS X25.000 Y0.000 Z15.000
N120 M30	

По сравнению с примером 6 здесь параметр <CorComp> = 3, поэтому можно не указывать параметр <GeoAx>. Теперь содержащееся в _CORVAL[0] значение действует непосредственно на компонент инструмента L1, вращение в N80 не влияет на результат, компоненты износа в \$TC_DP12 передаются вместе с _CORVAL[0] в геометрический компонент, поэтому из-за \$TC_DP13 вся длина инструмента уже после первого вызова SETTCOR в N100 стоит в геометрическом компоненте инструмента.

Пример 8

Программный код	Комментарий
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; токарный инструмент
N30 \$TC_DP3[1,1]=10.0	; геометрия L1
N40 \$TC_DP4[1,1]=15.0	; геометрия L2
N50 \$TC_DP5[1,1]=20.0	; геометрия L3
N60 \$TC_DP12[1,1]=10.0	; износ L1
N70 \$TC_DP13[1,1]=0.0	; износ L2
N80 \$TC_DP14[1,1]=0.0	; износ L3
N90 \$SC_WEAR_SIGN=TRUE	
N100 _CORVAL[0]=10.0	
N110 _CORVAL[1]=15.0	
N120 _CORVAL[2]=5.0	
N130 ROT Y-30	
N140 T1 D1 G18 G0	
N150 R1=SETTCOR(_CORVAL,"W",1,1)	
N160 T1 D1 X0 Y0 Z0	; ==> позиция MCS X7.990 Y25.000 Z31.160
N170 M30	

В N90 устанавливается SD42930 \$SC_WEAR_SIGN, т. е. износ должен быть нормирован отрицательным знаком. Коррекция является векторной (<CorComp> = 1), и вектор коррекции необходимо прибавить к износу (<CorMode> = 1). Геометрические отношения в плоскости Z-X представлены на следующем изображении:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <значение>

<Bit>	<значение>	Объяснение
0	0	В случае токарного и шлифовального инструмента параметр износа поперечной оси учитывается при вычислении значения радиуса :
	1	В случае токарного и шлифовального инструмента параметр износа поперечной оси учитывается при вычислении как значение диаметра :
1	0	В случае токарного и шлифовального инструмента компонент длины инструмента поперечной оси учитывается при вычислении как значение радиуса :
	1	В случае токарного и шлифовального инструмента компонент длины инструмента поперечной оси учитывается при вычислении как значение диаметра :

Если соответствующие биты установлены, то соответствующая строка нормируется с коэффициентом 0.5. Коррекция посредством SETTCOR выполняется таким образом, что все эффективное изменение длины инструмента равно указанному в <CorVal> значению. Если при расчете длины длина нормируется на основании машинных данных MD20360 \$MC_TOOL_PARAMETER_DEF_MASK с коэффициентом 0,5, то коррекция этого компонента должна выполняться с двойным указанным значением.

Пример

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK = 2 (длина инструмента должна быть нормирована в оси диаметра с коэффициентом 0.5)

Ось X – ось диаметра.

Программный код	Комментарий
N10 DEF REAL _LEN[11]	
N20 DEF REAL _CORVAL[3]	
N30 \$TC_DP1[1,1]=500	; Тип инструмента
N40 \$TC_DP2[1,1]=2	; Положение резцов
N50 \$TC_DP3[1,1]=3.	; Геометрия – длина 1
N60 \$TC_DP4[1,1]=4.	; Геометрия – длина 2
N70 \$TC_DP5[1,1]=5.	; Геометрия – длина 3
N80 _CORVAL[0]=1.	
N90 _CORVAL[1]=1.	
N100 _CORVAL[2]=1.	
N110 T1 D1 G18 G0 X0 Y0 Z0	; ==> позиция MCS X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL, "G", 1, 1)	
N130 T1 D1 X0 Y0 Z0	; ==> позиция MCS X2.5 Y6 Z5
N140 R3=\$TC_DP3[1,1]	; = 5. = (3.000 + 2.*1.000)
N150 R4=\$TC_DP4[1,1]	; = 5. = (4.000 + 1.000)
N160 R5=\$TC_DP5[1,1]	; = 6. = (5.000 + 1.000)
N170 M30	

Коррекция длины инструмента должна составлять 1 мм в каждой оси (с N80 по N100). Поэтому в длинах L2 и L3 к исходной длине прибавляется 1 мм. И наоборот, к исходной длине инструмента в L1 прибавляется двойное значение коррекции (2 мм), чтобы общая длина инструмента изменилась, как и требуется, на 1 мм. При сравнении с позициями подвода в кадрах N110 и N130 видно, что каждая позиция оси изменилась на 1 мм.

11.13 Считывание соотношения длин инструмента L1, L2, L3 с осями координат (LENTOAX)

Функция LENTOAX выводит информацию по соотношению длин инструмента L1, L2 и L3 **активного** инструмента с абсциссой, ординатой и аппликатой. Управление соотношением абсциссы, ординаты и аппликаты с гео-осями осуществляется через фреймы и активную плоскость (G17 – G19).

При этом рассматривается только геометрический компонент инструмента (от \$TC_DP3[<t>,<d>] до \$TC_DP5[<t>,<d>]), т. е. возможное отличие от данного соотношения осей других компонентов (например, износ) на результат не влияет.

Синтаксис

<состояние> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])

Принцип

LENTOAX (. . .):	Предопределенная функция выведения информации по соотносению длин инструмента L1, L2 и L3 активного инструмента с осями координат		
	Автономно в кадре:		да
<состояние>:	Возвращаемое значение функции. Отрицательные значения показывают состояния ошибки.		
	Тип данных:		INT
	Значение:	0	Функция ОК Информации в <AxInd> для описания достаточно (все компоненты длин инструмента параллельны гео-осям).
		1	Функция ОК, но для правильного описания необходимо обработать содержание <Matrix> (компоненты длин инструмента не параллельны гео-осям).
		-1	Недействительная строка в параметре <Coord>.
-2		нет активного инструмента.	
Параметры			

11.13 Считывание соотношения длин инструмента $L1$, $L2$, $L3$ с осями координат (LENTOAX)

1	<AxInd>:	Если компоненты длин инструмента параллельны гео-осям, то индексы осей, соотношенных с компонентами длин с L1 по L3, возвращаются в поле <AxInd>.			
		• <AxInd> [0]: Абсцисса • <AxInd> [1]: Ордината • <AxInd> [2]: Аппликата			
		Тип данных:	INT[3]		
		Значение:	<table><tr><td>0</td><td>Соотнесение отсутствует (ось не существует)</td></tr><tr><td>1 ... 3 или -1 ... -3</td><td>номер длины, действующей в соответствующей оси координат. Знак является отрицательным, если компонент длины инструмента указывает в отрицательном направлении координат.</td></tr></table>	0	Соотнесение отсутствует (ось не существует)
0	Соотнесение отсутствует (ось не существует)				
1 ... 3 или -1 ... -3	номер длины, действующей в соответствующей оси координат. Знак является отрицательным, если компонент длины инструмента указывает в отрицательном направлении координат.				
Если не все компоненты длин параллельны или встречно-параллельны гео-осям, то в <AxInd> возвращается индекс оси, содержащий самую большую долю компонента длин инструмента. В этом случае (если только функция не выводит ошибку по иной причине) возвращаемое значение <Status> = 1. Тогда отображение компонентов длин инструмента с L1 по L3 на гео-осях с 1-ой по 3-ю полностью описывается содержанием 2-го параметра <Matrix>.					
2	<Matrix>:	Матрица, отображающая вектор длин инструмента (L1=1, L2=1, L3=1) в вектор осей координат (абсцисса, ордината, аппликата), т. е. столбцам присвоены компоненты длин инструмента в последовательности L1, L2, L3, строкам — оси в последовательности: абсцисса, ордината, аппликата.			
		Тип данных:	REAL		
		В матрице всегда действительны все элементы, в т. ч., если относящаяся к оси координат гео-ось отсутствует, т. е. если соответствующий элемент в <AxInd> = 0.			
3	<Coord>:	Система координат, для которой действует соотнесение (опционально)			
		Тип данных:	STRING		
		Символ:	MCS M	Отображение длины инструмента в систему координат станка	
			BCS B	Отображение длины инструмента в базовую кинематическую систему	
			WCS W	Отображение длины инструмента в систему координат инструмента (по умолчанию)	
			KCS K	Отображение длины инструмента в систему координат инструмента кинематической трансформации	
			TCS T	Отображение длины инструмента в систему координат инструмента	
		Написание символов в строке (прописное или строчное) является произвольным.			
Если параметр <Coord> не указывается, используется WCS (по умолчанию).					

Примечание

В TCS всегда все компоненты длин инструмента параллельны или встречно-параллельны осям.

Встречно-параллельными компоненты могут быть только тогда, когда активно отражение и установлены следующие установочные данные:

SD42900 \$SC_MIRROR_TOOL_LENGTH (смена знака длины инструмента при отражении)

Пример

Стандартный случай: фрезерный инструмент при G17.

L1 действует в Z (аппликата), L2 действует в Y (ордината), L3 действует в X (абсцисса).

Вызов функции в форме:

<состояние>=LENTOAX (<AxInd>, <Matrix>, "WCS")

В этом случае параметр результата <AxInd> содержит значения:

<AxInd>[0] = 3

<AxInd>[1] = 2

<AxInd>[2] = 1

или кратко: (3, 2, 1)

Соответствующая матрица (<Matrix>) в этом случае:

$$\langle \text{Matrix} \rangle = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Переключение с G17 на G18 или G19 ничего не меняет в результате, т. к. соотношение компонентов длин с гео-осями изменяется также, как и соотношение абсциссы, ординаты и аппликаты.

Теперь при активной G17 программируется вращение фрейма вокруг Z на 60 градусов, например,

ROT Z60

Направление аппликаты (направление Z) остается неизменным, основной компонент L2 теперь лежит в направлении новой оси X, основной компонент L1 в отрицательном направлении оси Y. Поэтому возвращаемое значение <Status> = 1, <AxInd> содержит значения (2, -3, 1).

Соответствующая матрица (<Matrix>) в этом случае:

$$\langle \text{Matrix} \rangle = \begin{pmatrix} 0 & \sin 60^\circ & \cos 60^\circ \\ 0 & \cos 60^\circ & -\sin 60^\circ \\ 1 & 0 & 0 \end{pmatrix}$$

Параметры траектории

12.1 Тангенциальное управление

12.1.1 Определение соединения (TANG)

С помощью предварительно определенной процедуры TANG(...) определяется тангенциальное соединение между круговой осью в качестве ведомой оси и двумя геометрическими осями в качестве ведущих осей. При этом ведомая ось непрерывно выравнивается относительно касательной к траектории ведущей оси.

Примечание

коэффициент связи

Явного программирования коэффициента связи 1 не требуется.
Коэффициент связи -1 поворачивает направление тангенциальной оси.

Синтаксис

TANG (<ведомая ось>, <ведущая ось_1>, <ведущая ось_2>, <коэффициент связи>, <система координат>, <оптимизация>)

Объяснение

TANG (. . .) :	Определение тангенциального соединения	
<ведомая ось>:	Имя ведомой оси (круговая ось)	
	Тип данных:	AXIS
	Диапазон значений:	Имя оси канала
<ведущая ось_1> <ведущая ось_2>	Имена ведущих осей (геометрических осей) ¹⁾	
	Тип данных:	AXIS
	Диапазон значений:	Имена геометрических осей канала
<коэффициент связи>:	Коэффициент n изменения угла ведомой оси относительно изменения касательной к траектории ведущих осей: Изменение угла $\text{ведомая ось} = \text{изменение угла}_{\text{касательная к траектории}} * n$	
	Тип данных:	REAL
	Значение по умолчанию:	1,0

<Система координат>:	Действующая система координат ²⁾		
	Тип данных:	CHAR	
	Значение:	"B":	Базовая система координат (значение по умолчанию)
		"W":	Система координат детали (недоступно)
<Оптимизация>:	Вид оптимизации		
	Тип данных:	CHAR	
	Значение:	"S":	Стандарт (значение по умолчанию) Динамика круговой оси не оказывает обратного действия на ведущие оси. Если динамика круговой оси превышает значение, необходимое для слежения, это перемещение отличается достаточной точностью. Если динамика круговой оси недостаточно высока для отслеживания изменения касательной к траектории, центровка круговой оси вдоль неопределенного пути перешлифовки будет отличаться от заданной центровки.
		"P":	Динамика круговой оси учитывается при планировании траектории ведущих осей. Для этого при включении тангенциального соединения командой TANGON() необходимо указать два дополнительных параметра: • Путь перешлифовки • Угловой допуск См. главу "Включение соединения (TANGON) (Страница 536)" Примечание В связи с кинематическими трансформациями рекомендуется использовать вид оптимизации "P".

Указание

Значения по умолчанию не требуют явного программирования.

¹⁾ Примечание

В качестве ведущих осей тангенциального соединения следует использовать геометрические оси, которые относительно исходного положения станка проходят запрограммированную траекторию в системе координат станка (MCS). Если, например, на фрезеровальном станке с качающейся головкой применяется поворотный цикл CYCLE800, то, в зависимости от конфигурации цикла, происходит интерполяция в WKS, например, по геометрическим осям X и Y. Тангенциальное соединение должно быть, напротив, определено с геометрическими осями в качестве ведущих осей, которые проходят в WCS запрограммированную траекторию. Для этого в качестве ведущих осей следует использовать геометрические оси в **неповернутом** состоянии станка.

²⁾ Примечание

Базовая система координат (BKS), в отличие от MCS, не может быть повернута. Если, например, через команду ROT или цикл поворота CYCLE800 БСК поворачивается, то тангенциальное управление перестает работать корректно.

12.1.2 Включение создания промежуточного кадра (TLIFT)

Если изменение касательной ведомой оси в какой-либо точке запрограммированной траектории ведущих осей превышает предельное значение, заданное в машинных данных MD37400 \$MA_EPS_TLIFT_TANG_STEP, то дальнейшее планирование траектории будет зависеть от настроенных параметров угла. Без использования предварительно определенной процедуры TLIFT(...) перемещение будет осуществляться в соответствии с методом перешлифовки, заданным в TANG(...) (Страница 533) и TANGON(...) (Страница 536).

Включение создания промежуточного кадра

При программировании TLIFT(...) в связи с TANG(...) в ходе предварительной обработки при распознавании угла в этом месте траектории вводится промежуточный кадр, автоматически созданный системой управления.

При выполнении программы по достижении промежуточного кадра ведущие оси приостанавливаются. В рамках промежуточного кадра ведомая ось вращается с максимальной динамикой в направлении касательной к траектории следующего кадра. После этого ведущие оси перемещаются далее по запрограммированной траектории.

Отключение создания промежуточного кадра

Для выключения создания промежуточного кадра необходимо переопределить тангенциальное соединение посредством TANG(...), но без последующего включения создания промежуточного кадра при помощи TLIFT(...).

Синтаксис

TLIFT (<ведомая ось>)

Значение

TLIFT (...):	Включение распознавания углов с расчетом промежуточного кадра	
<ведомая ось>:	Имя ведомой оси (круговая ось)	
	Тип данных:	AXIS
	Диапазон значений:	Имя оси канала

Скорость вращения ведомой оси

Траекторная ось

Если ведомая ось перед активацией тангенциального соединения уже перемещалась в качестве траекторной оси, вращение в промежуточном кадре происходит в качестве траекторной оси.

При задании исходного радиуса при помощи FGREF [<ось>]=0.001 вращение осуществляется с параметрированной максимальной осевой скоростью:

MD32000 \$MA_MAX_AX_VELO[<ведомая ось>]

Позиционирующая ось

Если ведомая ось перед активацией тангенциального соединения еще не перемещалась в качестве траекторной оси, вращение в промежуточном кадре происходит в качестве позиционирующей оси.

При этом вращение осуществляется с параметрированной скоростью позиционирующей оси:

MD32060 \$MA_POS_AX_VELO[<ведомая ось>]

12.1.3 Включение соединения (TANGON)

Предварительно определенная процедура TANGON(...) включает определенное ранее в TANG(...) (Страница 533) тангенциальное соединение. При этом ведомая ось непрерывно выравнивается относительно касательной к траектории ведущей оси при последующем перемещении ведущих осей.

Угол ведомой оси

Угол, занимаемый ведомой осью относительно касательной к траектории, зависит от заданного в TANG(...) коэффициента трансформации, заданного в машинных данных MD37402 \$MA_TANG_OFFSET угла смещения и действующего дополнительного к нему угла смещения, заданного через TANGON(...).

Оптимизация «Р»

Если при определении тангенциального соединения (TANG(...)) в качестве параметра оптимизации будет указано значение «Р», то при включении соединения потребуется указать параметр «Путь перешлифовки» и, возможно, параметр «Угловой допуск».

Если в качестве углового допуска будет задано значение 0, действует только параметр «Путь перешлифовки».

Если в качестве углового допуска будет задано значение выше 0, то эффективный путь перешлифовки будет складываться из минимального параметрированного пути перешлифовки и пути перешлифовки на основании параметрированного углового допуска.

Если динамика ведомой оси недостаточна для отслеживания параметрированных условий, то скорость движения ведущих осей по траектории будет соответственно снижена.

Синтаксис

TANGON (<ведомая ось>, <угол смещения>, <путь перешлифовки>, <угловой допуск>)

Значение

TANGON (. . .) :	Включение тангенциального соединения	
<ведомая ось>:	Имя ведомой оси (круговая ось)	
	Тип данных:	AXIS
	Диапазон значений:	Имя оси канала
<Угол смещения>:	Угол смещения ведомой оси относительно касательной к траектории Исходной точкой служит нулевая точка круговой оси.	
	Тип данных:	REAL
<Путь перешлифовки>:	Макс. допустимый путь перешлифовки Если путь перешлифовки вследствие динамических условий окажется больше, то скорость движения ведущих осей по траектории будет снижена.	
	Тип данных:	REAL
<угловой допуск>:	Макс. допустимый допуск относительно заданного угла между нулевым положением ведомой оси и касательной к траектории	
	Тип данных:	REAL

12.1.4 Выключение соединения (TANGOF)

Предварительно определенная процедура TANGOF(...) выключает тангенциальное соединение, определенное в TANG(...) (Страница 533) и включенное в TANGON(...) (Страница 536). После этого ведомая ось прекращает выравниваться по касательной к траектории ведущей оси. Соединение ведомой оси с ведущими осями сохраняется даже после выключения, что препятствует использованию, например, следующих функций:

- Смена плоскостей
- Переключение геометрических осей
- Определение нового тангенциального соединения для ведомой оси

Полный разрыв соединения ведомой и ведущих осей происходит только после удаления соединения посредством TANGDEL(...) (Страница 538).

Программирование

TANGOF (<ведомая ось>)

Значение

TANGOF (. . .) :	Выключение тангенциального соединения	
<ведомая ось>:	Имя ведомой оси (круговая ось)	
	Тип данных:	AXIS
	Диапазон значений:	Имя оси канала

12.1.5 Удаление соединения (TANGDEL)

Тангенциальное соединение, определенное в TANG(...) (Страница 533), сохраняется даже после отключения тангенциального соединения в TANGOF(...) (Страница 537). Существующее тангенциальное соединение препятствует выполнению, например, следующих пунктов:

- Смена плоскостей
- Переключение геометрических осей
- Определение нового тангенциального соединения для ведомой оси

Предварительно определенная процедура TANGDEL(...) после выключения тангенциального соединения в TANGOF(...) удаляет имеющее тангенциальное соединение.

Синтаксис

TANGDEL (<ведомая ось>)

Значение

TANGDEL (. . .) :	Удаление тангенциального соединения, определенного в TANG()	
	Активность:	Покадрово
<ведомая ось>:	Имя ведомой оси, тангенциальное соединение которой нужно удалить	
	Тип данных:	AXIS
	Диапазон значений:	Имя оси канала

Примеры

Смена ведущей оси

Перед тем, как определить для ведомой оси новое тангенциальное соединение с другой ведущей осью, потребуется предварительно удалить имеющееся тангенциальное соединение.

Программный код	Комментарий
N10 TANG (A, X, Y, 1)	; определение тангенциального соединения для ведомой оси A: A к X и Y
N20 TANGON (A)	; включение тангенциального соединения для ведомой оси A
N30 X10 Y20	
...	
N80 TANGOF (A)	; выключение тангенциального соединения для ведомой оси A
N90 TANGDEL (A)	; удаление тангенциального соединения для ведомой оси A
...	
N120 TANG (A, X, Z)	; определение нового тангенциального соединения для ведомой оси A
N130 TANGON (A)	; включение нового тангенциального соединения для ведомой оси A

Программный код	Комментарий
...	

Переключение геометрических осей

Перед переключением геометрической оси для имеющегося соединения необходимо предварительно удалить соединение.

Программный код	Комментарий
N10 GEOAX (2, Y1)	; 2. Геометрическая ось = ось станка Y1
N20 TANG (A, X, Y)	; определение тангенциального соединения для ведомой оси A
N30 TANGON (A, 90)	; включение тангенциального соединения для ведомой оси A
N40 G2 F8000 X0 Y0 I0 J50	; кадр движения
N50 TANGOF (A)	; выключение тангенциального соединения для ведомой оси A
N60 TANGDEL (A)	; удаление тангенциального соединения для ведомой оси A
N70 GEOAX (2, Y2)	; 2. Геометрическая ось = ось станка Y2
N80 TANG (A, X, Y)	; определение нового тангенциального соединения для ведомой оси A
N90 TANGON (A, 90)	; включение нового тангенциального соединения для ведомой оси A
...	

12.2 Характеристика подачи (FNORM, FLIN, FCUB, FPO)

Для более гибкой задачи характеристики подачи программирование подачи по DIN 66025 дополнено линейными и кубическими характеристиками.

Кубические характеристики могут программироваться напрямую или как интерполирующие сплайны. Благодаря этому возможно – в зависимости от изгиба обрабатываемой детали – непрерывное программирование ровных характеристик скоростей.

Эти характеристики скорости обеспечивают изменения ускорения без рывков и тем самым изготовление равномерных поверхностей детали.

Синтаксис

F... FNORM
F... FLIN
F... FCUB
F=FPO (... , ... , ...)

Значение

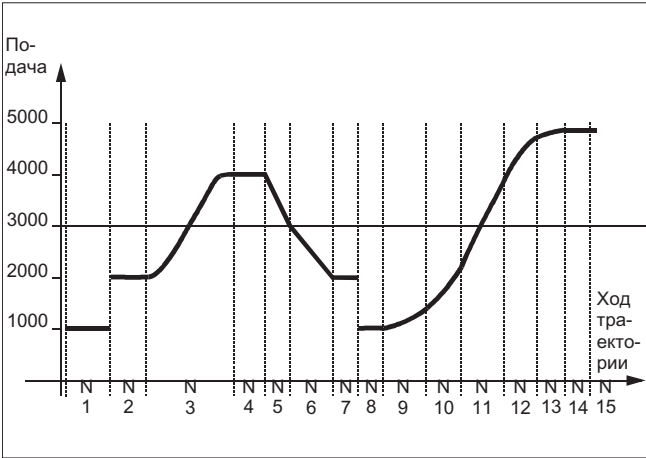
FNORM:	Первичная установка. Значение подачи задается через ход траектории кадра и после действует как модальное значение.
FLIN:	Профиль скорости движения по траектории линейный : Значение подачи линейно выводится от актуального значения в начале кадра до конца кадра через ход траектории и после действует как модальное значение. Такое поведение может комбинироваться с G93 и G94.
FCUB:	Профиль скорости движения по траектории кубический : Запрограммированные покадрово значения F соединяются - относительно конечной точки кадра - через сплайн. Сплайн начинается и заканчивается тангенциально к предшествующим или последующим данным подачи и действует с G93 и G94. Если адрес F отсутствует в кадре, то для этого используется последнее запрограммированное значение F.
F=FPO... :	Профиль скорости движения по траектории через полином : Адрес F обозначает характеристику подачи через полином от актуального значения до конца кадра. После конечное значение считается модальным значением.

Оптимизация подачи на изогнутых участках траектории

Полином подачи F=FPO и сплайн подачи FCUB всегда должны проходиться с постоянной скоростью резания CFC . Благодаря этому возможно создание постоянного по ускорению профиля заданной подачи.

Пример: Различные профили подачи

В этом примере показано программирование и графическое отображение различных профилей подачи.



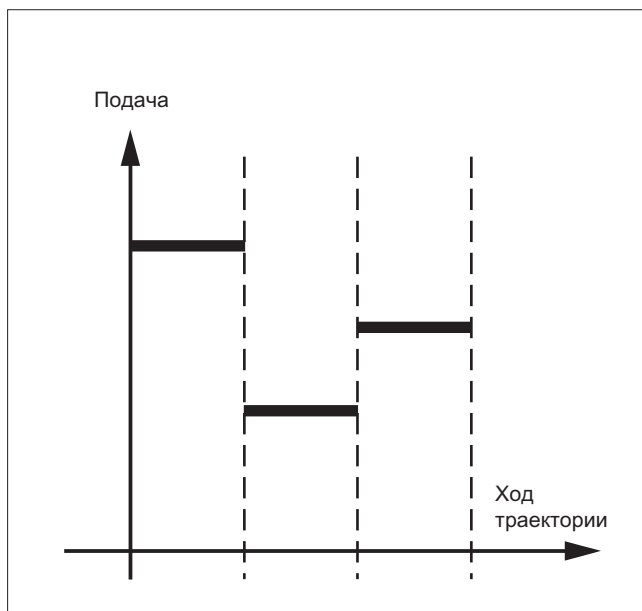
Программный код	Комментарий
N1 F1000 FNORM G1 X8 G91 G64	; Постоянный профиль подачи, указание составного размера
N2 F2000 X7	; Скачкообразное изменение заданной скорости
N3 F=FPO(4000, 6000, -4000)	; Профиль подачи через полином с подачей 4000 на конце кадра.
N4 X6	; Полиномиальная подача 4000 действует как модальное значение.
N5 F3000 FLIN X5	; Линейный профиль подачи
N6 F2000 X8	; Линейный профиль подачи
N7 X5	; Линейная подача действует как модальное значение
N8 F1000 FNORM X5	; Постоянный профиль подачи со скачкообразным изменением ускорения.
N9 F1400 FCUB X8	; Все последующие покадрово запрограммированные значения F соединяются сплайнами.
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	
N14 FNORM X5	; Отключить профиль сплайна.
N15 X20	

Дополнительная информация

FNORM

Адрес подачи F обозначает подачу по траектории как постоянное значение по DIN 66025.

Подробности см. руководство по программированию "Основы".

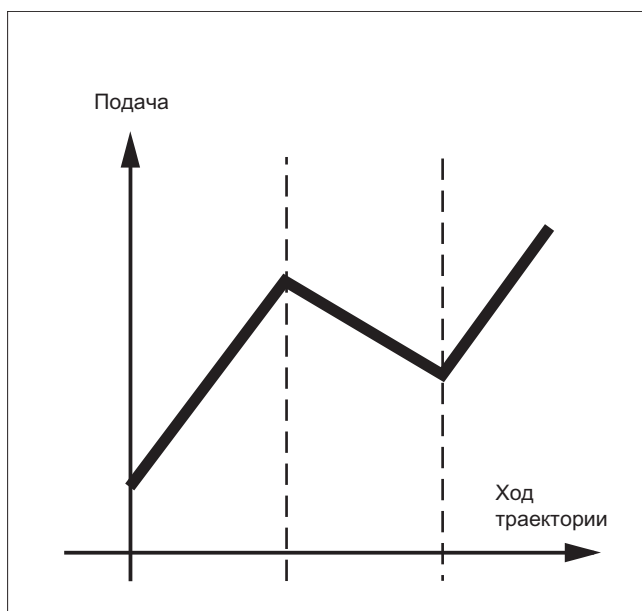


FLIN

Характеристика подачи выводится линейно от актуального значения подачи к запрограммированному значению F до конца кадра.

Пример:

```
N30 F1400 FLIN X50
```



FCUB

Подача выводится от актуального значения подачи к запрограммированному значению F до конца кадра по кубической характеристике. СЧПУ соединяет все

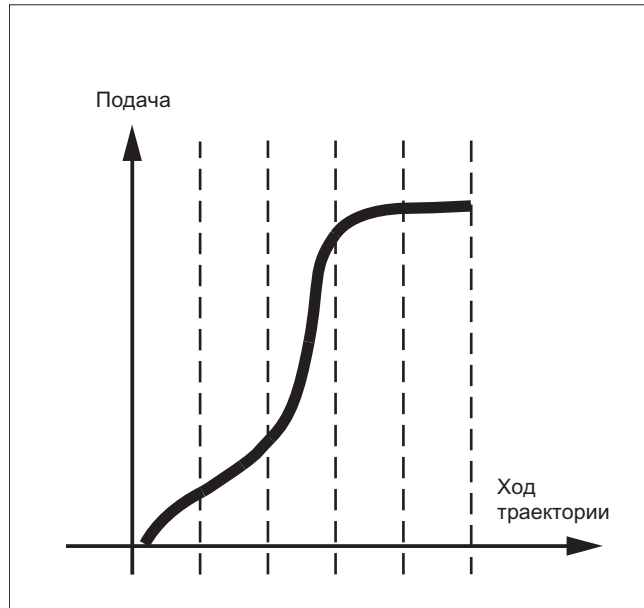
запрограммированные покадрово с активной FCUB значения подачи через сплайны. Значения подачи служат здесь как опорные точки для вычисления сплайн-интерполяции.

Пример:

N50 F1400 FCUB X50

N60 F2000 X47

N70 F3800 X52



F=FPO(...,...)

Характеристика подачи программируется напрямую через полином. Указание коэффициентов полинома выполняется аналогично полиномиальной интерполяции.

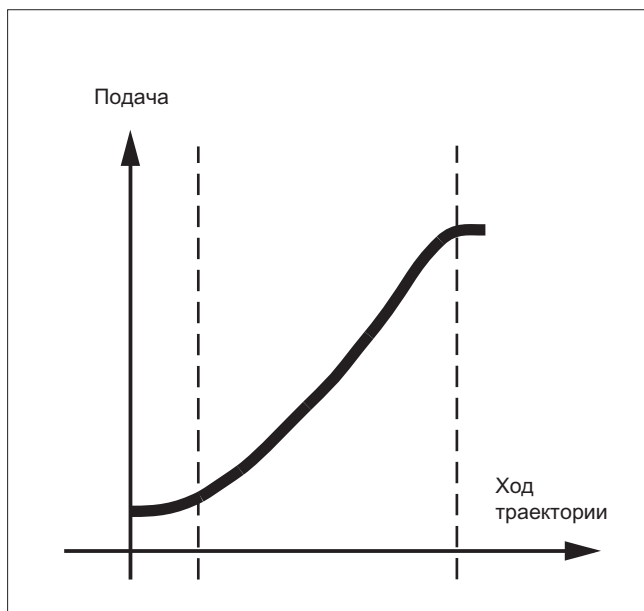
Пример:

F=FPO(endfeed, quadf, cubf)

endfeed, quadf и cubf это заранее определенные переменные.

endfeed:	подача на конце кадра
quadf:	квадратный коэффициент полинома
cubf:	кубический коэффициент полинома

При активной FCUBсплайн в начале и в конце кадра тангенциально примыкает к установленной через FPO характеристике.



Граничные условия

- Независимо от запрограммированной характеристики подачи действуют функции для программирования параметров движения по траектории.
- Запрограммированная характеристика подачи всегда действует абсолютно – независимо от G90 или G91.
- Характеристика подачи FLIN и FCUB действует с G93 и G94, **не действует** при G95, G96/G961 и G97/G971.
- При активном компрессоре COMPON и объединении нескольких кадров в сегмент сплайна действительно следующее:

FNORM:	Для сегмента сплайна действует слово F последнего соответствующего кадра.
FLIN:	Для сегмента сплайна действует слово F последнего соответствующего кадра. Запрограммированное значение F действует к концу сегмента и после выполняется линейный подвод.
FCUB:	Созданный сплайн подачи отклоняется макс. на определенное в машинных данных MD20172 \$MC_COMPRESS_VELO_TOL значение от запрограммированных конечных точек.
F=FPO (... F ... F ...):	Эти кадры не сжимаются.

12.3 Динамические характеристики

12.3.1 Режим ускорения (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)

Для программирования режима ускорения имеются следующие команды программы обработки детали:

- "BRISK, BRISKA"
Отдельные оси или траекторные оси движутся с макс. ускорением до достижения запрограммированной скорости подачи (**Ускорение без ограничения рывка**).
- "SOFT, SOFTA"
Отдельные оси или траекторные оси движутся с постоянным ускорением до достижения запрограммированной скорости подачи (**Ускорение с ограничением рывка**).
- "DRIVE, DRIVEA"
Отдельные оси или траекторные оси движутся с макс. ускорением до сконфигурированной границы скорости (установка MD!). После этого осуществляется уменьшение ускорения (установка MD!) до достижения запрограммированной скорости подачи.

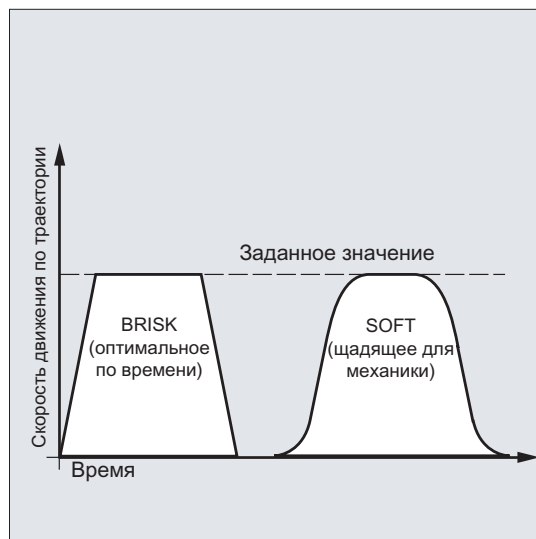


Рисунок 12-1 Характеристика скорости движения по траектории при BRISK и SOFT

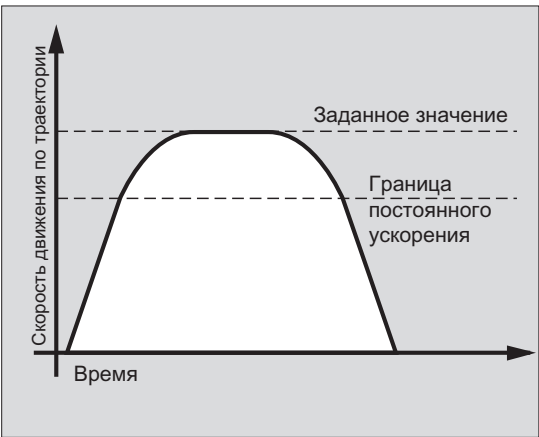


Рисунок 12-2 Характеристика скорости движения по траектории при DRIVE

Синтаксис

```
BRISK
BRISKA (<ось1>,<ось2>,...)
SOFT
SOFTA (<ось1>,<ось2>,...)
DRIVE
DRIVEA (<ось1>,<ось2>,...)
```

Значение

BRISK:	Команда для включения "Ускорения без ограничения рывка" для траекторных осей.
BRISKA:	Команда для включения "Ускорения без ограничения рывка" для движений отдельных осей (JOG, JOG/INC, позиционирующая ось, качающаяся ось, и т.д.).
SOFT:	Команда для включения "Ускорения с ограничением рывка" для траекторных осей.
SOFTA:	Команда для включения "Ускорения с ограничением рывка" для движений отдельных осей (JOG, JOG/INC, позиционирующая ось, качающаяся ось, и т.д.).
DRIVE:	Команда для включения уменьшенного ускорения выше сконфигурированной границы скорости (MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT) для траекторных осей.
DRIVEA:	Команда для включения уменьшенного ускорения выше сконфигурированной границы скорости (MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT) для движений отдельных осей (JOG, JOG/INC, позиционирующая ось, качающаяся ось, и т.д.).
(<ось1>,<ось2>,...):	Отдельные оси, для которых должен действовать вызванный режим ускорения.

Граничные условия

Смена режима ускорения при обработке

Если в программе обработки детали режим ускорения изменяется при обработке (BRISK ↔ SOFT), то и в режиме управления траекторией на переходе выполняется смена кадра с точным остановом на конце кадра.

Примеры

Пример 1: SOFT и BRISKA

Программный код

```
N10 G1 X... Y... F900 SOFT
N20 BRISKA (AX5,AX6)
...
```

Пример 2: DRIVE и DRIVEA

Программный код

```
N05 DRIVE
N10 G1 X... Y... F1000
N20 DRIVEA (AX4, AX6)
...
```

Литература

Описание функций "Основные функции"; Ускорение (B2)

12.3.2 Управление ускорением для ведомых осей (VELOLIMA, ACCLIMA, JERKLIMA)

В соединениях осей (тангенциальное слежение, буксировка, соединение по главному значению, электронный редуктор; см. "Соединения осей (Страница 583)") ведомые оси/шпиндели перемещаются в зависимости от одной или нескольких ведущих осей/шпинделей.

Управление динамическими ограничениями ведомых осей/шпинделей возможно с помощью функций VELOLIMA, ACCLIMA и JERKLIMA из программы обработки детали или из синхронных действий, и при уже активированном соединении осей.

Примечание

Функция JERKLIMA доступна не для всех типов соединений.

Литература:

- Описание функций "Специальные функции"; Соединения осей (M3)
- Описание функций "Расширенные функции"; Синхронный шпиндель (S3)

Примечание

Доступность для SINUMERIK 828D

Функции VELOLIMA, ACCLIMA и JERKLIMA могут использоваться на SINUMERIK 828D только в комбинации с функцией "Буксировка"!

Синтаксис

VELOLIMA (<ось>)=<значение>

ACCLIMA (<ось>)=<значение>

JERKLIMA (<ось>)=<значение>

Значение

VELOLIMA:	Команда для коррекции спараметрированной максимальной скорости
ACCLIMA:	Команда для коррекции спараметрированного максимального ускорения
JERKLIMA:	Команда для коррекции спараметрированного максимального рывка
<ось>:	Ведомая ось, динамические ограничения которой должны быть исправлены
<значение>:	Процентная поправка

Примеры

Пример 1: Коррекция динамических ограничений для ведомой оси (AX4)

Программный код	Комментарий
...	
VELOLIMA[AX4]=75	; Коррекция ограничения до 75 % зафиксированной в машинных данных макс. осевой скорости.
ACCLIMA[AX4]=50	; Коррекция ограничения до 50 % зафиксированного в машинных данных макс. осевого ускорения.
JERKLIMA[AX4]=50	; Коррекция ограничения до 50 % зафиксированного в машинных данных макс. осевого рывка при движении по траектории.
...	

Пример 2: Электронный редуктор

Ось 4 через соединение "Электронный редуктор" соединена с осью X. Приемистость ведомой оси ограничивается до 70 % макс. ускорения. Макс. допустимая скорость ограничивается до 50 % макс. скорости. После успешного включения соединения макс. допустимая скорость снова устанавливается на 100 %.

Программный код	Комментарий
...	
N120 ACCLIMA[AX4]=70	; Уменьшенное макс. ускорение.
N130 VELOLIMA[AX4]=50	; Уменьшенная макс. скорость.
...	
N150 EGON(AX4, "FINE", X, 1, 2)	; Включение соединения электронного редуктора.

Программный код	Комментарий
...	
N200 VELOLIMA[AX4]=100	; Полная макс. скорость.
...	

Пример 3: Управления соединением по главному значению через статическое синхронное действие

Ось 4 через соединение по главному значению соединяется с осью X. Динамическая характеристика через статическое синхронное действие 2 от позиции 100 ограничивается до 80 %.

Программный код	Комментарий
...	
N120 IDS=2 WHENEVER \$AA_IM[AX4] >	; Синхронное действие
100 DO ACCLIMA[AX4]=80	
N130 LEADON (AX4, X, 2)	; Соединение по главному значению вкл
...	

12.3.3 Активация спец. для технологии динамических значений (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)

С помощью G-группы "Технология" для 5 различных технологических этапов обработки можно активировать подходящую динамику.

Динамические значения и G-команды могут конфигурироваться и поэтому зависят от установок машинных данных (→ Изготовитель станка!).

Литература:

Описание функций "Основные функции"; Режим управления траекторией, точный останов, LookAhead (B1)

Синтаксис

Активация динамический значений:

DYNNORM
DYNPOS
DYNROUGH
DYNSEMIFIN
DYNFINISH

Примечание

Динамические значения активируются уже в том кадре, в котором программируется соответствующая G-команда. Обработка не останавливается.

Запись или чтение определенного элемента поля:

R<m>=\$MA... [n, X]
\$MA... [n, X]=<значение>

Значение

DYNNORM:	G-команда для активации обычной динамики		
DYNPOS:	G-команда для активации динамики для режима позиционирования, нарезания внутренней резьбы		
DYNROUGH:	G-команда для активации динамики для черновой обработки		
DYNSEMIFIN:	G-команда для активации динамики для чистовой обработки		
DYNFINISH:	G-команда для активации динамики для точной чистовой обработки		
R<m>:	R-параметр с номером <m>		
\$MA... [n, X]:	Машинные данные с определяющим динамику элементом поля		
<n>:	Индекс поля		
	Диапазон значений:		0 ... 4
	0	Обычная динамика (DYNNORM)	
	1	Динамика для режима позиционирования (DYNPOS)	
	2	Динамика для черновой обработки (DYNROUGH)	
	3	Динамика для чистовой обработки (DYNSEMIFIN)	
	4	Динамика для точной чистовой обработки (DYNFINISH)	
<X>:	Адрес оси		
<значение>:	Динамическое значение		

Примеры

Пример 1: Активация динамических значений

Программный код	Комментарий
DYNNORM G1 X10	; Первичная установка
DYNPOS G1 X10 Y20 Z30 F...	; Режим позиционирования, нарезание внутренней резьбы
DYNROUGH G1 X10 Y20 Z30 F10000	; Черновая обработка
DYNSEMIFIN G1 X10 Y20 Z30 F2000	; Чистовая обработка
DYNFINISH G1 X10 Y20 Z30 F1000	; Точная чистовая обработка

Пример 2: Запись или чтение определенного элемента поля

Макс. ускорение для черновой обработки, ось X.

Программный код	Комментарий
R1=\$MA_MAX_AX_ACCEL[2,X]	; Чтение
\$MA_MAX_AX_ACCEL[2,X]=5	; Запись

12.4 Движение с предупреждением (FFWON, FFWOF)

Благодаря предупреждению зависящий от скорости путь выбега уменьшается практически до нуля. Движение с предупреждением способствует более высокой точности контура и тем самым лучшим производственным результатам.

Синтаксис

FFWON

FFWOF

Значение

FFWON:	Команда для включения предупреждения
FFWOF:	Команда для выключения предупреждения

Примечание

Через машинные данные устанавливается тип предупреждения и то, какие траекторные оси должны перемещаться через предупреждение.

Стандарт: зависящее от скорости предупреждение

Опция: зависящее от ускорения предупреждение

Пример

Программный код
N10 FFWON
N20 G1 X... Y... F900 SOFT

12.5 Программируемая точность контура (CPRECON, CPRECOF)

Функция "Программируемая точность контура" сокращает погрешности траектории на изогнутых контурах за счет автоматической адаптации скорости.

Соблюдаемая точность контура задается в зависимости от конфигурации станка (MD20470 \$MC_MC_CPREC_WITH_FFW; см. данные изготовителя станка) либо через установочные данные \$SC_CONTPREC, либо через запрограммированный допуск контура CTOL. Чем меньше значение и чем меньше коэффициент K_v гео-осей, тем сильнее уменьшается подача по траектории на изогнутых контурах.

Функция "Программируемая точность контура" включается или выключается через операторы CPRECON и CPRECOF в программе ЧПУ.

Синтаксис

CPRECON

...

CPRECOF

Объяснение

CPRECON:	Вызов G-команды: "Программируемая точность контура" включить	
	Активность:	модально
CPRECOF:	Вызов G-команды: "Программируемая точность контура" выключить	
	Активность:	модально

CPRECON и CPRECOF вместе образуют группу G-функций 39 (программируемая точность контура).

Примечание

Через установочные данные \$SC_MINFEED (мин. подача по траектории при CPRECON) пользователь может задать мин. скорость для подачи по траектории.

Подача не ограничивается этим значением, разве что было запрограммировано более низкое F-значение или динамические ограничения осей вызывают более низкую скорость движения по траектории.

Примечание

Функция "Программируемая точность контура" рассматривает только гео-оси контура. Она не влияет на скорости позиционирующих осей.

Пример

Программный код	Комментарий
N10 G0 X0 Y0	
N20 CPRECON	; Включение "Программируемой точности контура".

12.5 Программируемая точность контура (CPRECON, CPRECOF)

Программный код	Комментарий
N30 G1 G64 X100 F10000	; Обработка с 10 м/мин в режиме управления траекторией.
N40 G3 Y20 J10	; Автоматическое ограничение подачи в круговом кадре.
N50 G1 X0	; Подача снова без ограничения (10 м/мин).
...	
N100 CPRECOF	; Выключение "Программируемой точности контура".
N110 G0 ...	

Литература

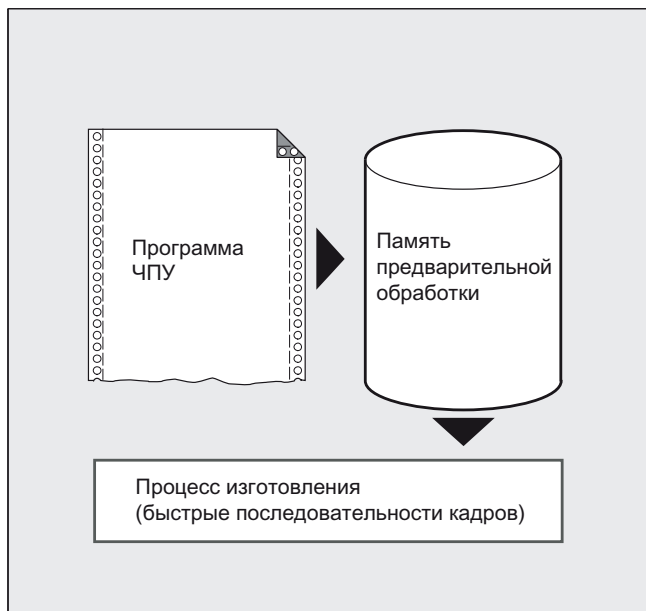
По программированию CTOL см. "Программирование допуска контура/ориентации (CTOL, OTOL, ATOL) (Страница 576)"

Более подробную информацию по функции "Программируемая точность контура" можно найти в:

Описание функций "Специальные функции"; Контроль туннеля контура (K6), глава: "Программируемая точность контура"

12.6 Выполнение программы с памятью предварительной обработки (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

В зависимости от конфигурации, СЧПУ имеет определенное число памяти предварительной обработки, которые сохраняют подготовленные кадры перед выполнением и выводят их в процессе производства как быстрые последовательности кадров. Благодаря этому возможно прохождение коротких путей с высокими скоростями. Если это допускает остаточное время СЧПУ, память предварительной обработки всегда заполняется.



Обозначить участок обработки

Участок обработки, который должен быть буферизирован в память предварительной обработки, обозначается в программе обработки детали в начале с "STOPFIFO" и на конце с "STARTFIFO". Выполнение подготовленных и буферизированных кадров начинается только после команды "STARTFIFO" или заполнения памяти предварительной обработки.

Автоматическое управление памятью предварительной обработки

Автоматическое управление памятью предварительной обработки вызывается командой "FIFOCTRL". "FIFOCTRL" сначала действует идентично "STOPFIFO". При каждом программировании ожидается заполнение памяти предварительной обработки и после начинается выполнение. Разница состоит в поведении при опустошении памяти предварительной: С "FIFOCTRL" начиная с уровня заполнения в 2/3 скорость движения по траектории значительно уменьшается, чтобы не допустить полного опустошения и торможения до состояния покоя.

Остановка предварительной обработки

Подготовка и буферизация кадров останавливается, если в кадре запрограммирована команда "STOPRE". Следующий кадр выполняется только тогда, когда все

12.6 Выполнение программы с памятью предварительной обработки (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

подготовленные и сохраненные прежде кадры полностью выполнены. Предыдущий кадр останавливается с точным остановом (как G9).

ВНИМАНИЕ

Отмена программы

При включенной коррекция инструмента и при сплайн-интерполяциях нельзя программировать "STOPRE", т. к. в этом случае будут прерваны связанные последовательности кадров.

Синтаксис

Таблица 12-1 Обозначить участок обработки:

```
STOPFIFO
...
STARTFIFO
```

Таблица 12-2 Автоматическое управление памятью предварительной обработки:

```
...
FIFOCTRL
...
```

Таблица 12-3 Остановка предварительной обработки:

```
...
stopre
...
```

Примечание

Команды "STOPFIFO", "STARTFIFO", "FIFOCTRL" и "STOPRE" должны программироваться в отдельном кадре.

12.6 Выполнение программы с памятью предварительной обработки (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

Значение

STOPFIFO:	"STOPFIFO" обозначает начало участка обработки, который должен быть помещен в память предварительной обработки. С "STOPFIFO" обработка останавливается и память предварительной обработки заполняется: - до определения "STARTFIFO" или "STOPRE" или - до заполнения памяти предварительной обработки или - до достижения конца программы.
STARTFIFO:	С "STARTFIFO" запускается быстрое выполнение участка обработки, параллельно с этим осуществляется заполнение памяти предварительной обработки
FIFOCTRL:	Включение автоматического управления памятью предварительной обработки
STOPRE:	Остановить предварительную обработку

Примечание

Заполнение памяти предварительной обработки не выполняется или прерывается, если участок обработки содержит команды, вызывающие режим без буферизации (реферирование, измерительные функции, ...).

Примечание

При обращении к данным состояния станка (\$SA...), СЧПУ создает внутреннюю остановку предварительной обработки.

Пример: Остановить предварительную обработку

Программный код	Комментарий
...	
N30 MEAW=1 G1 F1000 X100 Y100 Z50	; Кадр измерения со щупом первого измерительного входа и линейная интерполяция.
N40 STOPRE	; Остановка предварительной обработки.
...	

12.7 Определение областей задержки останова (DELAYFSTON, DELAYFSTOF)

Для определения условно прерываемой области задержки останова в программе обработки детали служат предварительно определенные процедуры DELAYFSTON и DELAYFSTOF.

Примечание

В синхронных действиях использование DELAYFSTON и DELAYFSTOF **не** допускается!

Синтаксис

```
DELAYFSTON
...
DELAYFSTOF
```

Объяснение

DELAYFSTON:	Определение начала области задержки останова	
	Автономно в кадре:	да
DELAYFSTOF:	Определить конец области задержки останова	
	Автономно в кадре:	да

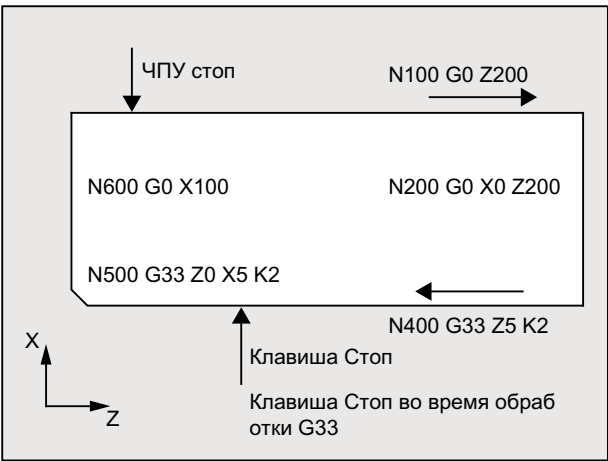
Пример программирования

Циклично повторяется следующий программный блок:

Программный код

```
...
N99 MY_LOOP:
N100 G0 Z200
N200 G0 X0 Z200
N300 DELAYFSTON
N400 G33 Z5 K2 M3 S1000
N500 G33 Z0 X5 K3
N600 G0 X100
N700 DELAYFSTOF
N800 GOTOB MY_LOOP
...
```

На следующем изображении видно, что пользователь нажимает на Стоп в области задержки останова, а ЧПУ начинает процесс торможения вне области задержки останова, т. е. в кадре N100. Тем самым ЧПУ останавливается в передней области N100.



Дополнительная информация

Конец подпрограммы

При завершении подпрограммы, в которой была вызвана DELAYFSTON, не явно активируется DELAYFSTOF.

Вложенность

Если подпрограмм 1 в области задержки останова вызывает подпрограмму 2, то вся подпрограмма 2 это область задержки останова. В частности, DELAYFSTOF не действует в подпрограмме 2.

Пример:

Программный код	Комментарий
N10010 DELAYFSTON	; Кадры с N10xxx программный уровень 1.
N10020 R1 = R1 + 1	
N10030 G4 F1	; Область задержки останова начинается.
...	
N10040 подпрограмма2	
...	
...	; Интерпретация подпрограммы 2.
N20010 DELAYFSTON	; Не действует, повторное начало, 2-й уровень.
...	
N20020 DELAYFSTOF	; Не действует, завершение на другом уровне.
N20030 RET	
N10050 DELAYFSTOF	; Конец области задержки останова в той же плоскости.
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	; Область задержки останова завершается. Остановы действуют с этого места сразу же.

12.7 Определение областей задержки останова (DELAYFSTON, DELAYFSTOF)

Системные переменные

О нахождении этапа программы обработки детали в данный момент времени в области задержки останова можно судить по следующим системным переменным:

- в программе обработки детали - \$P_DELAYFST
- в синхронных действиях с \$AC_DELAYFST

Значение	Объяснение
0	Область задержки останова не активна
1	Область задержки останова активна

12.8 Пропуск места в программе для SERUPRO (IPTRLOCK, IPTRUNLOCK)

Для определенных сложных механических ситуаций на станке необходимо недопущение поиска кадра SERUPRO.

С помощью программируемого указателя прерываний существует возможность вмешательства для установа при "Поиске на месте прерывания" перед непригодным для поиска местом.

Могут быть также определены непригодные для поиска области в областях программы обработки детали, повторный вход ЧПУ в которые невозможен. При отмене программы ЧПУ отмечает последний обработанный кадр, поиск которого возможен через интерфейс пользователя HMI.

Синтаксис

IPTRLOCK
IPTRUNLOCK

Команды стоят отдельно в одной строке программы обработки детали и обеспечивают возможность программируемого указателя прерываний.

Объяснение

IPTRLOCK:	Начало непригодного для поиска сегмента программы
IPTRUNLOCK:	Конец непригодного для поиска сегмента программы

Обе команды допускаются только в программах обработки детали, но **не** в синхронных действиях.

Пример

Вложение непригодных для поиска сегментов программы на двух программных уровнях с не явной "IPTRUNLOCK". Не явная "IPTRUNLOCK" в подпрограмме 1 завершает непригодную для поиска область.

Программный код	Комментарий
N10010 IPTRLOCK ()	
N10020 R1 = R1 + 1	
N10030 G4 F1	; Кадр останова, начинающий непригодный для поиска сегмент программы.
...	
N10040 подпрограмма2	
...	; Интерпретация подпрограммы 2.
N20010 IPTRLOCK ()	; Не действует, повторное начало.
...	
N20020 IPTRUNLOCK ()	; Не действует, завершение на другом уровне.
N20030 RET	
...	
N10060 R2 = R2 + 2	

12.8 Пропуск места в программе для SERUPRO (IPTRLOCK, IPTRUNLOCK)

Программный код	Комментарий
N10070 RET	; Конец непригодного для поиска сегмента программы.
N100 G4 F2	; Главная программа продолжается.

После прерывание на 100 снова выводит указатель прерываний.

Дополнительная информация

Регистрация и поиск непригодных для поиска областей

Непригодные для поиска сегменты программы обозначаются с помощью языковых команд "IPTRLOCK" и "IPTRUNLOCK".

Команда "IPTRLOCK" фиксирует указатель прерываний на исполняемом в главном ходе отдельном кадре (SBL1). Этот кадр в дальнейшем обозначается как кадр останова. Если после "IPTRLOCK" возникает отмена программы, то на интерфейсе пользователя HMI можно искать этот так называемый кадр останова.

Повторный установ на актуальный кадр

Указатель прерываний с помощью "IPTRUNLOCK" для следующего сегмента программы будет установлен на актуальный на момент прерывания кадр.

После найденной цели поиска с тем же кадром останова можно повторить новую цель поиска.

Отредактированный пользователем указатель прерываний снова должен быть удален через HMI.

Правила вложенности

Следующие моменты регулируют взаимодействие языковых команд "IPTRLOCK" и "IPTRUNLOCK" со вложенностями и концом программы обработки детали:

1. При завершении подпрограммы, в которой была вызвана "IPTRLOCK", , не явно активируется "IPTRUNLOCK".
2. "IPTRLOCK" в непригодной для поиска области не действует.
3. Если подпрограмма 1 вызывает в непригодной для поиска области подпрограмму 2, то вся подпрограмма 2 остается непригодной для поиска. В частности, "IPTRUNLOCK" не действует в подпрограмме 2.

Дополнительную информацию см.

/FB1/ Описание функций "Основные функции"; GPP, канал, программный режим (K1).

Системная переменная

Непригодная для поиска область может быть распознана в программе обработки детали с помощью "\$P_IPTRLOCK".

Автоматический указатель прерываний

Функция автоматического указателя прерываний автоматически устанавливается определенным ранее тип связи как непригодный для поиска. Посредством машинных данных автоматический указатель прерываний активируется для

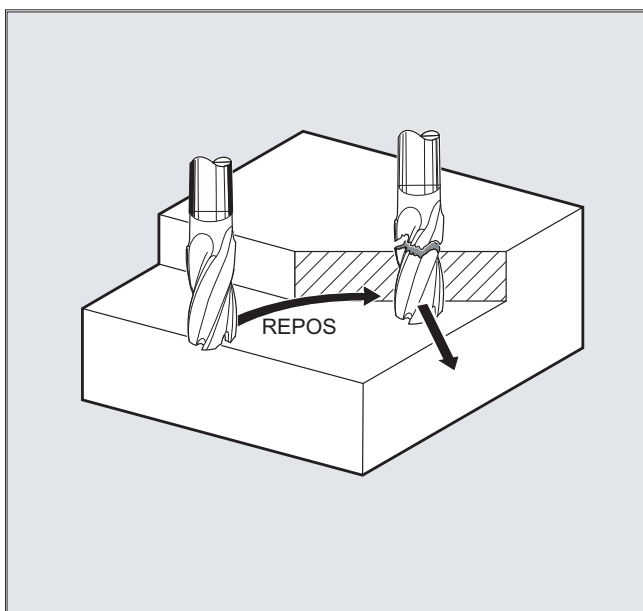
- электронного редуктора при "EGON",
- осевого соединения по главному значению при "LEADON".

При пересечении запрограммированного и активируемого через машинные данные автоматического указателя прерываний, создается макс. возможная непригодная для поиска область.

12.9 Повторный подвод к контуру (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Если при обработке текущая программа прерывается и выполняется отвод инструмента – к примеру, из-за поломки инструмента или для измерения детали – то затем можно выполнить повторный подвод к контуру в выбранной точке, используя программное управление.

Команда `REPOS` действует в ASUP, как возврат из подпрограммы (например, M17). Последующие кадры больше не исполняются. По прерыванию выполнения программы см. также "Обработчик прерываний (ASUP) (Страница 134)".



Синтаксис

```
REPOSA RMIBL DISPR=...
REPOSA RMBBL
REPOSA RMEBL
REPOSA RMNBL
REPOSL RMIBL DISPR=...
REPOSL RMBBL
REPOSL RMEBL
REPOSL RMNBL
REPOSQ RMIBL DISPR=... DISR=...
REPOSQ RMBBL DISR=...
REPOSQ RMEBL DISR=...
REPOSQA DISR=...
REPOSH RMIBL DISPR=... DISR=...
REPOSH RMBBL DISR=...
REPOSH RMEBL DISR=...
REPOSHA DISR=...
```

Значение

Выбор пути подвода

REPOSA:	Повторный подвод к контуру с геометрическими осями на одной прямой. Все прочие оси канала также повторно позиционируются.
REPOSL:	Повторный подвод к контуру с геометрическими осями на одной прямой. Все прочие оси канала должны быть явно запрограммированы.
REPOSQ DISR=... :	Повторный подвод к контуру с геометрическими осями на четверти окружности с радиусом DISR. Все прочие оси канала должны быть явно запрограммированы.
REPOSQA DISR=... :	Повторный подвод к контуру с геометрическими осями на четверти окружности с радиусом DISR. Все прочие оси канала также повторно позиционируются.
REPOSH DISR=... :	Повторный подвод к контуру с геометрическими осями на полуокружности с диаметром DISR. Все прочие оси канала должны быть явно запрограммированы.
REPOSHA DISR=... :	Повторный подвод к контуру с геометрическими осями на полуокружности с радиусом DISR. Все прочие оси канала также повторно позиционируются.

Выбор точки повторного подвода

RMIBL:	Подвод к точке прерывания
RMIBL DISPR=...:	Точка входа на расстоянии DISPR в мм/дюймах перед точкой прерывания
RMBBL:	Подвод к начальной точке кадра
RMEBL:	Подвод к конечной точке кадра
RMEBL DISPR=... :	Подвод к конечной точке кадра на расстоянии DISPR перед конечной точкой
RMNBL:	Подвод к ближайшей точке траектории
A0 B0 C0 :	Оси, в которых должен выполняться подвод

Примечание

Совместимость

Для обеспечения совместимости с прежними версиями ПО, можно продолжать программировать режим подвода REPOS через модальные команды RMI, RMB, RME и RMN. При использовании внутри ASUP эту подпрограмму нужно снабдить атрибутом SAVE в операторе PROC. В противном случае используемый в ASUP модальный режим подвода REPOS, если он отклоняется от предустановки RMI, действует и для последующих процессов REPOS.

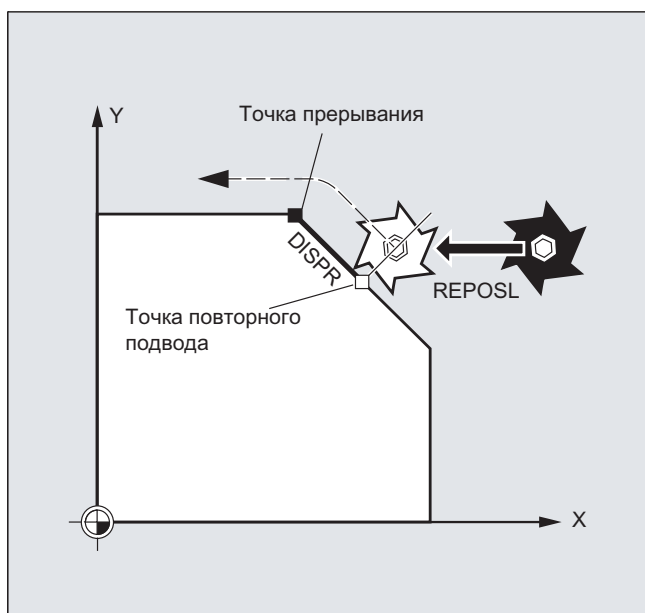
Повторный подвод к контуру на прямой, REPOSA, REPOSL

Инструмент движется к точке повторного подвода по прямой.

12.9 Повторный подвод к контуру (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Пример

REPOSL RMIBL DISPR=6 F400

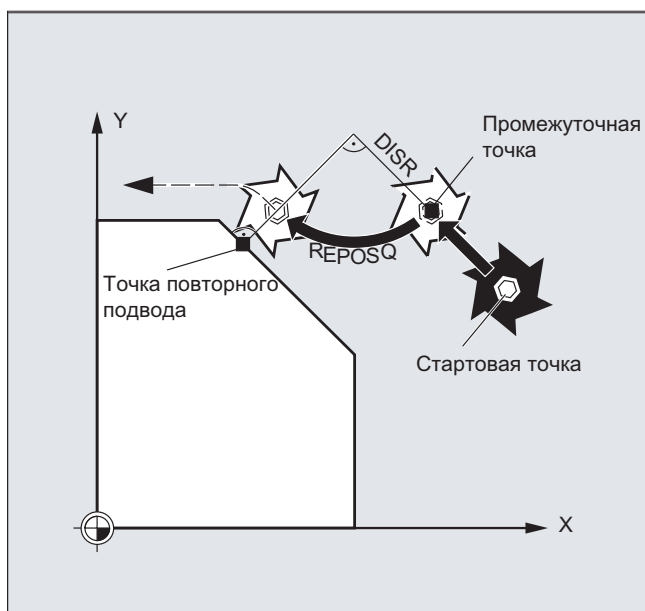


Повторный подвод к контуру по четверти окружности, REPOSQ, REPOSQA

Инструмент движется к точке повторного подвода по четверти круга с радиусом $DISR = \dots$. Необходимую промежуточную точку между стартовой точкой и точкой повторного подвода СЧПУ вычисляет автоматически.

Пример

REPOSQ RMIBL DISR=10 F400

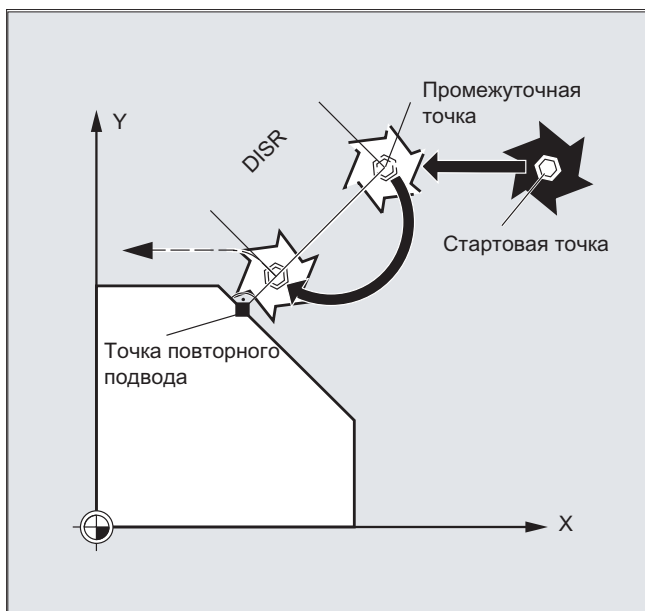


Повторный подвод к контуру по полуокругу, REPOSH, REPOSHA

Инструмент движется к точке повторного подвода по полуокругу с диаметром DISR=
Необходимую промежуточную точку между стартовой точкой и точкой повторного подвода СЧПУ вычисляет автоматически.

Пример

REPOSH RMIBL DISR=20 F400

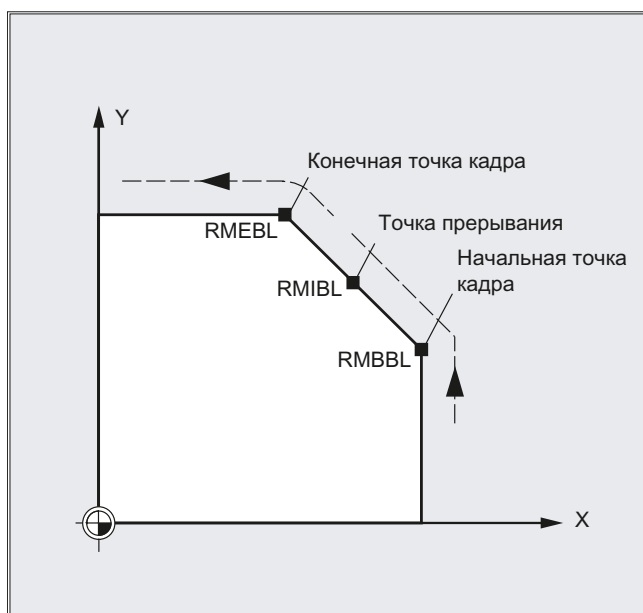


Определение точки повторного подвода (не для подвода SERUPRO с RMNBL)

Относительно кадра ЧПУ, в котором выполнение программы было прервано, можно выбирать между тремя точками повторного подвода:

- RMIBL, точка прерывания
- RMBBL, начальная точка кадра или последняя конечная точка
- RMEBL, конечная точка кадра

12.9 Повторный подвод к контуру (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)



С RMIBL DISPR=... или с RMEBL DISPR=... можно определить точку повторного подвода, лежащую перед точкой прерывания или перед конечной точкой кадра.

С DISPR=... описывается путь контура в мм/дюймах, на который точка повторного подвода удалена от точки прерывания или конечной точки. Эта точка может и для больших по величине значений лежать макс. в начальной точке кадра.

Если DISPR=... не программируется, то действует DISPR=0 и тем самым точка прерывания (для RMIBL) или конечная точка кадра (для RMEBL).

Знак DISPR

Знак DISPR обрабатывается. При положительном знаке сохраняется прежнее поведение.

При отрицательном знаке продолжение за точкой прерывания или при RMBBL за стартовой точкой.

Расстояние м/у точкой прерывания и точкой возобновления получается из величины DISPR. Эта точка может и для больших по величине значений лежать макс. в конечной точке кадра.

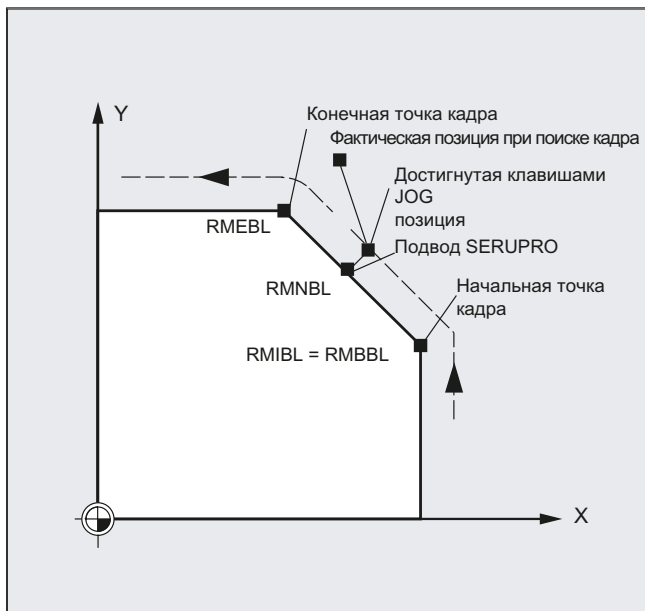
Пример использования:

Датчик определяет приближение к прихвату. Запускается ASUP, с которой выполняется обход прихвата.

После с отрицательной DISPR выполняется репозиционирование на точку за прихватом и программа продолжается.

Подвод SERUPRO с RMNBL

Если при обработке в любом месте инициируется отмена, то при подводе SERUPRO с RMNBL выполняется переход кратчайшего пути от места отмены, чтобы после лишь обработать остаточный путь. Для этого пользователь запускает процесс SERUPRO на кадр прерывания и выполняет позиционирование с помощью клавиш JOG на позицию перед поврежденным местом конечного кадра.



Примечание

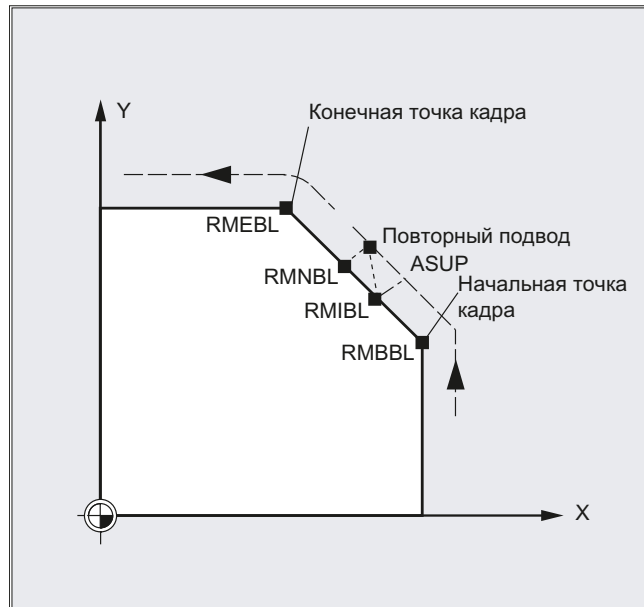
SERUPRO

Для SERUPRO RMIBL и RMBBL идентичны. RMNBL ограничивается не только SERUPRO, но имеет общее действие.

12.9 Повторный подвод к контуру (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Подвод к ближайшей точке траектории RMNBL

На момент интерполяции REPOSA после прерывания кадр повторного подвода с RMNBL не начинается заново, а лишь обрабатывается остаточный путь. Выполняется подвод к ближайшей точке траектории прерванного кадра.



Состояние для действительного режима REPOS

Действительный режим REPOS прерванного кадра может быть считан через синхронные действия с переменной \$AC_REPOS_PATH_MODE:

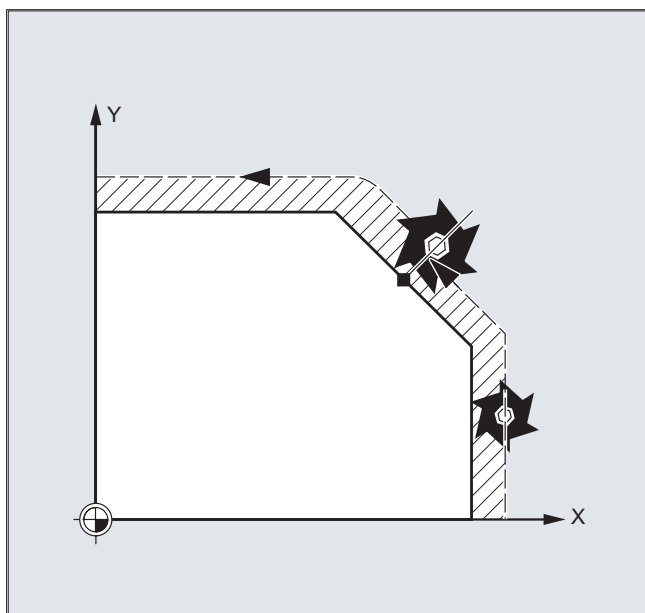
- 0 подвод не определен
- 1 RMBBL: подвод к началу
- 2 RMIBL: подвод к точке прерывания
- 3 RMEBL: подвод к конечной точке кадра
- 4 RMNBL: подвод к ближайшей точке траектории прерванного кадра

Подвод с новым инструментом

Если выполнение программы было остановлено из-за поломки инструмента:

При программировании нового номера D программа продолжается с точки повторного подвода с измененными значениями коррекции инструмента.

При изменении значений коррекции инструмента может случиться, что подвод к точке прерывания будет более невозможен. В этом случае выполняется подвод к ближайшей от точки прерывания точке на новом контуре (возможно измененной на DISPR).



Подвод к контуру

Движение, с которым инструмент снова подводится к контуру, может быть запрограммировано. Адреса перемещаемых осей указываются со значением ноль.

С помощью команд REPOSA, REPOSQA и REPOSHA выполняется автоматическое репозиционирование всех осей. Указание оси не требуется.

При программировании REPOSL, REPOSQ и REPOSH все гео-оси выполняют подвод автоматически, т. е. и без указания в команде. Все другие оси должны быть указаны в команде.

Для круговых движений REPOSH и REPOSQ действует:

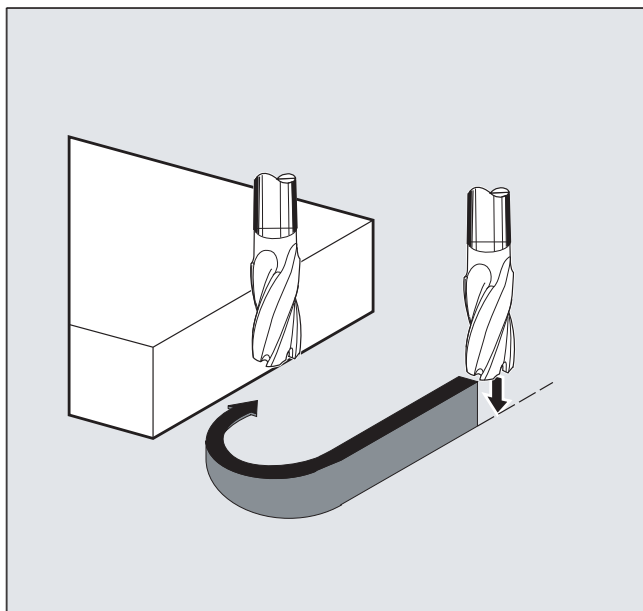
Окружность проходится в указанной рабочей плоскости G17 до G19.

Если в кадре подвода указывается третья гео-ось (направление подачи), то подвод к точке повторного подвода в том случае, когда позиция инструмента и запрограммированная позиция в направлении подачи не совпадают, выполняется по спирали.

12.9 Повторный подвод к контуру (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

В следующих случаях происходит автоматическое переключение на линейный подвод REPOSL:

- Значение для DISR не указано.
- Определенного направления подвода не существует (прерывание программы в кадре без информации перемещения).
- При направлении подвода вертикально к актуальной рабочей плоскости.



12.10 Воздействие на управление движением

12.10.1 Процентная коррекция рывка (JERKLIM)

С помощью команды ЧПУ "JERKLIM" установленный через машинные данные макс. возможный рывок оси при движении по траектории в критических сегментах программы может быть уменьшен или увеличен.

Условие

Режим ускорения SOFT должен быть активен.

Активность

Функция действует:

- в автоматических режимах работы.
- только на траекторные оси.

Синтаксис

JERKLIM[<ось>]=<значение>

Значение

JERKLIM:	Команда для коррекции рывка	
<ось>:	Ось станка, предельное значение рывка которой должно быть согласовано.	
<значение>:	Процентная поправка, относящаяся к сконфигурированному макс. рывку оси при движении по траектории (MD32431 \$MA_MAX_AX_JERK).	
	Диапазон значений:	1 ... 200
	Значение 100 не вызывает управления рывком.	

Примечание

Поведение JERKLIM при завершении программы обработки детали и сбросе канала конфигурируется с битом 0 в машинных данных MD32320 \$MA_DYN_LIMIT_RESET_MASK:

- Бит 0 = 0:
Запрограммированное значение для JERKLIM при Reset канала/M30 сбрасывается на 100 %.
- Бит 0 = 1:
Запрограммированное значение для JERKLIM сохраняется после Reset канала/M30.

Пример

Программный код	Комментарий
...	
N60 JERKLIM[X]=75	; Осевые салазки в направлении X должны ускоряться/замедляться только макс. с 75 % от допустимого для оси рывка.
...	

12.10.2 Процентная коррекция скорости (VELOLIM)

С помощью команды VELOLIM в программе обработки детали или синхронном действии можно уменьшить установленную через машинные данные макс. возможную скорость оси или макс. возможную, зависящую от ступени редуктора частоту вращения шпинделя.

Активность

Функция действует:

- в автоматических режимах работы.
- на траекторные и позиционирующие оси.
- на шпиндели в шпиндельном/осевом режиме

Синтаксис

VELOLIM[<ось/шпиндель>]=<значение>

Значение

VELOLIM:	Команда для коррекции скорости
<ось/шпиндель>:	Ось или шпиндель, предельное значение скорости или частоты вращения которых должно быть согласовано. VELOLIM для шпинделей Через машинные данные (MD30455 \$MA_MISC_FUNCTION_MASK, бит 6) для программирования в программе обработки детали можно установить, действует ли "VELOLIM" независимо от актуального использования как шпиндель или ось (бит 6 = 1) или должна программироваться отдельно для каждого режима работы (бит 6 = 0). Если сконфигурировано раздельное действие, то выбор осуществляется через идентификатор при программировании: • Идентификатор шпинделя S<n> для режимов работы шпинделя • Идентификатор оси, например "C", для осевого режима

<значение>:	Процентная поправка Поправка относится:	
	- для осей / шпинделей в осевом режиме (MD30455 бит 6 == 0): к сконфигурированной макс. скорости оси (MD32000 \$MA_MAX_AX_VELO). - для шпинделей в шпиндельном или осевом режиме (MD30455 бит 6 == 1): к макс. частоте вращения активной ступени редуктора оборотов (MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])	
	Диапазон значений:	1 ... 100
Значение 100 не вызывает управления скоростью или частотой вращения.		

Примечание

Поведение при завершении программы обработки детали и сбросе канала

Поведение "VELOLIM" при завершении программы обработки детали и сбросе канала может быть установлено через машинные данные: MD32320
\$MA_DYN_LIMIT_RESET_MASK, бит 0

Определение активного ограничения частоты вращения в шпиндельном режиме

Ограничение частоты вращения через "VELOLIM" (меньше 100 %) в шпиндельном режиме может быть определено через следующую системную переменную:

- \$AC_SMAXVELO (макс. возможная частота вращения шпинделя)
- \$AC_SMAXVELO_INFO (идентификатор для причины ограничения частоты вращения)

Примеры

Пример 1: Ограничение скорости оси станка

Программный код	Комментарий
...	
N70 VELOLIM[X]=80	; Осевые салазки в направлении X должны перемещаться только макс. с 80 % от допустимой для оси скорости.
...	

Пример 2: Ограничение частоты вращения шпинделя

Программный код	Комментарий
N05 VELOLIM[S1]=90	; Ограничение макс. частоты вращения шпинделя 1 до 90 % от 1000 об/мин.
...	
N50 VELOLIM[C]=45	; Ограничение частоты вращения до 45 % от 1000 об/мин, C это идентификатор оси S1.

Программный код	Комментарий
...	

Установки машинных данных для шпинделя 1 (AX5)

- Макс. частота вращения ступени редуктора 1 = 1000 об/мин:
MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
- Программирование "VELOLIM" действует совместно для шпиндельного и осевого режима, независимо от запрограммированного идентификатора:
MD30455 \$MA_MISC_FUNCTION_MASK[AX5], бит 6 = 1

12.10.3 Пример программы для JERKLIM и VELOLIM

Программа ниже представляет пример использования для процентного ограничения рывка и скорости:

Программный код	Комментарий
N1000 G0 X0 Y0 F10000 SOFT G64	
N1100 G1 X20 RNDM=5 ACC[X]=20 ACC[Y]=30	
N1200 G1 Y20 VELOLIM[X]=5	; Осевые салазки в направлении X должны перемещаться только с 5 % от допустимой для оси скорости.
JERKLIM[Y]=200	; Осевые салазки в направлении Y должны ускоряться/замедляться макс. с 200 % от допустимого для оси рывка.
N1300 G1 X0 JERKLIM[X]=2	; Осевые салазки в направлении X должны ускоряться/замедляться макс. с 2 % от допустимого для оси рывка.
N1400 G1 Y0	
M30	

12.11 Программирование допуска контура/ориентации (CTOL, OTOL, ATOL)

Адресами CTOL, OTOL, ATOL можно подбирать определенные в машинных и установочных данных допуски на обработку для функций компрессора, перешлифовку и сглаживание ориентации в программе обработки детали.

Запрограммированные значения допуска действуют до их повторного программирования или удаления через присваивание отрицательного значения. Кроме этого, они удаляются при завершении программы или сбросе (Reset). После удаления снова действуют установленные в параметрах значения допусков.

Синтаксис

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

Объяснение

CTOL:	Адрес программирования допуска контура		
	Сфера применения:	• всех функций компрессора • всех типов перешлифовки кроме G641 и G644	
	Остановка предварительной обработки:	нет	
	Активность:	модально	
	<Value>:	Значение для допуска контура это линейные данные.	
		Тип:	REAL
		Единица:	дюйм/мм (в зависимости от текущей настройки указания размеров)
		Диапазон значений:	≥ 0: Значение допуска
			< 0: Удаление запрограммированного значения допуска ⇒ Снова действует значение допуска, установленное в параметрах машинных или установочных данных.

12.11 Программирование допуска контура/ориентации (CTOL, OTOL, ATOL)

OTOL:	Адрес программирования допуска ориентации			
	Сфера применения:	• всех функций компрессора • сглаживания ориентации ORISON • всех типов перешлифовки, кроме G641, G644, OSD		
	Остановка предвари- тельной обработки:	нет		
	Активность:	модально		
	<Value>:	Значение для допуска ориентации это угловые данные.		
		Тип:	REAL	
		Единица:	Градус	
		Диапазон зна- чений:	≥ 0:	Значение допуска
			< 0:	Удаление запрограммированного зна- чения допуска ⇒ Снова действует значение допуска, установленное в параметрах машин- ных или установочных данных.
	ATOL:	Адрес программирования спец. для оси допуска		
Сфера применения:		• всех функций компрессора • сглаживания ориентации ORISON • всех типов перешлифовки, кроме G641, G644, OSD		
Остановка предвари- тельной обработки:		нет		
Активность:		модально		
<Axis>:		Имя оси канала, для которой должен действовать запрограм- мированный допуск		
<Value>:		Значение для осевого допуска, в зависимости от типа оси (ли- нейная или круговая ось) это линейные или угловые данные.		
		Тип:	REAL	
		Единица:	для линейных ос- ей:	дюйм/мм (в зависимости от текущей настройки указания размеров)
			для круговых ос- ей:	Градус
		Диапазон зна- чений:	≥ 0:	Значение допуска
			< 0:	Удаление запрограммированного зна- чения допуска ⇒ Снова действует значение допуска, установленное в параметрах машин- ных или установочных данных.

Примечание

Приоритет запрограммированных посредством CTOL и OTOL спец. для канала значений допуска выше, чем у запрограммированных посредством ATOL спец. для оси значений допуска.

Примечание

Масштабирующие фреймы

Масштабирующие фреймы действуют на запрограммированные допуски так же, как и на позиции осей, т. е. относительный допуск остается тем же.

Пример

Программный код	Комментарий
COMPCAD G645 G1 F10000	; Активировать функцию компрессора COMPCAD .
X... Y... Z...	; Здесь действуют машинные и установочные данные.
X... Y... Z...	
X... Y... Z...	
CTOL=0.02	; С этого места действует допуск контура в 0,02 мм.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
ASCALE X0.25 Y0.25 Z0.25	; С этого места действует допуск контура в 0,005 мм.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
CTOL=-1	; С этого места снова действуют машинные и установочные данные.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Системные переменные

Чтение с остановкой предварительной обработки

Через следующие системные переменные в программе обработки детали и синхронном действии считываются действующие на соответствующий момент допуски:

- **\$AC_CTOL**
Спец. для канала допуск контура, который действовал при подготовке текущего кадра главного хода
Если действующего допуска контура нет, то \$AC_CTOL выводит корень из суммы квадратов допусков гео-осей.
- **\$AC_OTOL**
Спец. для канала допуск ориентации, который действовал при подготовке текущего кадра главного хода.
Если действующего допуска ориентации нет, то \$AC_OTOL выводит во время текущей трансформации ориентации корень от суммы квадратов допусков осей ориентации, в иных случаях — значение "-1".
- **\$AA_ATOL[<ось>]**
Спец. для оси допуск контура, который действовал при подготовке текущего кадра главного хода.
Если допуск контура активен, то \$AA_ATOL[<гео-ось>] выводит допуск контура, деленный на корень из количества гео-осей.
Если допуск ориентации и трансформация ориентации активны, то \$AA_ATOL[<ось ориентации>] выводит допуск ориентации, деленный на корень из количества осей ориентации.

Примечание

Если значения допуска не были запрограммированы, то переменные \$A не являются достаточно дифференцированными для различения допусков отдельных функций.

Такие случаи возможны, если машинные и установочные данные устанавливают различные допуски для функций компрессора, перешлифовки и сглаживания ориентации. В этом случае системные переменные выводят наибольшее значение, встречающееся в активных на настоящий момент функциях. Если, например, активны функция компрессора с допуском ориентации 0,1° и сглаживание ориентации ORISON с 1°, то \$AC_OTOL выводит значение "1". Если сглаживание ориентации отключается, то \$AC_OTOL выводит значение "0,1"

Чтение без остановки предварительной обработки

Через следующие системные переменные в программе обработки детали считываются действующие на соответствующий момент допуски:

- **\$P_CTOL**
Действующий на соответствующий момент спец. для канала допуск контура
- **\$P_OTOL**
Действующий на соответствующий момент спец. для канала допуск ориентации
- **\$PA_ATOL**
Действующий на соответствующий момент спец. для оси допуск контура

Граничные условия

Запрограммированные посредством CTOL, OTOL, ATOL допуски действуют и для функций, косвенно зависящих от этих допусков:

- Ограничение ошибки хорды при расчете заданного значения
- Основные функции режима поверхностей произвольной формы

Программирование CTOL, OTOL, ATOL **не** затрагивает следующие функции перешлифовки:

- перешлифовка ориентации с OSD
OSD использует не допуск, а расстояние до перехода кадра.
- перешлифовка с G644
G644 служит не для обработки, а для оптимизации смен инструмента и других движений без обработки.
- перешлифовка с G645
G645 практически всегда ведет себя как G642 и тем самым использует запрограммированные допуски. Только на переходах (с постоянным касанием) между кадрами с перепадом кривизны, например, на тангенциальном переходе окружность — прямая, используется значение допуска из машинных данных MD33120 \$MA_PATH_TRANS_POS_TOL. Путь перешлифовки в таких местах может проходить и по внешней стороне запрограммированного контура, что не в полной мере поддерживается многими приложениями. Кроме этого, как правило, для компенсации изменений кривизны достаточно небольшого постоянного допуска, который не является проблемой для программиста ЧПУ.

12.12 Параметры смены кадров при активном соединении (CPBC)

С помощью команды CPBC задается критерий смены кадров, который должен быть выполнен, чтобы в программе обработки детали при активном соединении была выполнена смена кадров.

Синтаксис

CPBC [<ведомая ось>] = <критерий>

Объяснение

CPBC:	Критерий смены кадров при активном соединении	
<ведомая ось>:	идентификатор ведомой оси	
<критерий>:	критерий смены кадра	
	Тип:	STRING
	Значение	Значение: смена кадра выполняется
	"NOC"	независимо от состояния соединения
	"IPOSTOP"	при синхронном ходе со стороны заданного значения
	"COARSE"	при синхронном ходе со стороны фактического значения "грубом"
	"FINE"	при синхронном ходе со стороны фактического значения "точном"

Пример

Программный код

```
; смена кадра выполняется при:
; - соединение с ведомой осью X2 == активно
; - синхронный ход со стороны заданного значения == активен
CPBC[X2]="IPOSTOP"
```


Соединения осей

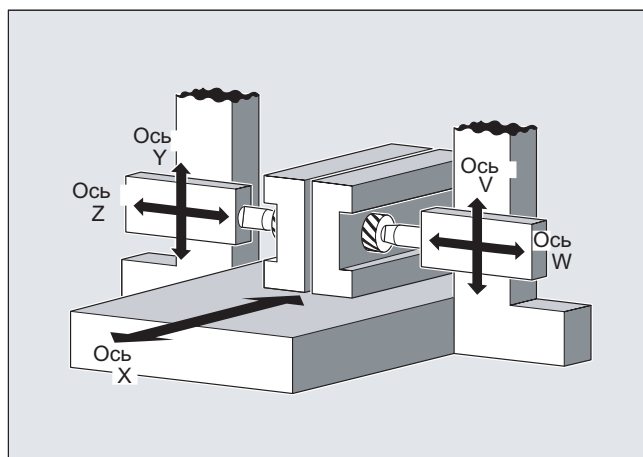
13.1 Буксировка (TRAILON, TRAILOF)

При движении определенной ведущей оси согласованные с ней буксируемые оси (= ведомые оси) проходят, с учетом коэффициента связи, произведенные ведущей осью пути перемещения.

Ведущая ось и ведомые оси вместе образуют структуру буксировок.

Области применения

- Перемещение оси через симулированную ось. Ведущая ось это симулированная ось, а буксируемая ось это реальная ось. Тем самым реальная ось может перемещаться с учетом коэффициента связи.
- Двухсторонняя обработка с 2 структурами буксировок:
 1. Ведущая ось Y, буксируемая ось V
 2. Ведущая ось Z, буксируемая ось W



Синтаксис

TRAILON (<ведомая ось>,<ведущая ось>,<коэффициент связи>)

TRAILOF (<ведомая ось>,<ведущая ось>,<ведущая ось 2>)

TRAILOF (<ведомая ось>)

Значение

TRAILON:	Команда для включения и определения структуры буксировок	
	Активность:	модально

13.1 Буксировка (TRAILON, TRAILOF)

<ведомая ось>:	Параметр 1: Обозначение буксируемой оси Примечание: Буксируемая ось также может быть ведущей осью для других буксируемых осей. Таким образом могут создаваться различные структуры буксировок.
<ведущая ось>:	Параметр 2: Обозначение ведущей оси
<коэффициент связи>:	Параметр 3: коэффициент связи Коэффициент связи указывает желаемое соотношение ходов буксируемой оси и ведущей оси: $\text{<коэффициент связи>} = \text{ход буксируемой оси} / \text{ход ведущей оси}$
	Тип: REAL
	Предустановка: 1
	Ввод отрицательного значения вызывает движение перемещения ведущей и буксируемой оси в противоположном направлении. Если коэффициент связи при программировании не указывается, то автоматически действует коэффициент связи 1.
TRAILOF:	Команда отключения структуры буксировок Активность: модально TRAILOF с 2 параметрами отключает только связь с указанной ведущей осью: <code>TRAILOF (<ведомая ось>, <ведущая ось>)</code> Если буксируемая ось имеет 2 ведущие оси, то для отключения обеих связей можно вызвать TRAILOF с 3 параметрами: <code>TRAILOF (<ведомая ось>, <ведущая ось>, <ведущая ось 2>)</code> Идентичный результат дает программирование TRAILOF без указания ведущей оси: <code>TRAILOF (<ведомая ось>)</code>

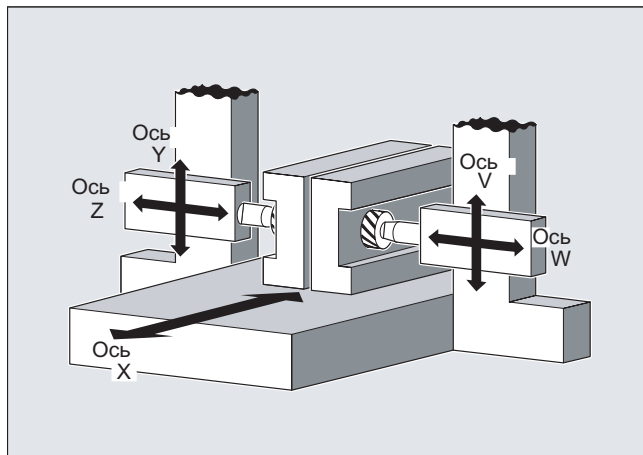
Примечание

Буксировка всегда выполняется в базовой кинематической системе (BKS).

Число активируемых одновременно структур буксировок ограничивается только возможностями комбинирования имеющихся на станке осей.

Пример

Деталь должна быть обработана с двух сторон с помощью представленной конфигурации осей. Для этого создается 2 структуры буксировок.



Программный код	Комментарий
...	
N100 TRAILON(V,Y)	; Включение 1-й структуры буксировок
N110 TRAILON(W,Z,-1)	; Включение 2-й структуры буксировок. Отрицательный коэффициент связи: Буксируемая ось движется в направлении, противоположном ведущей оси.
N120 G0 Z10	; Подача оси Z и W в противоположном осевом направлении.
N130 G0 Y20	; Подача оси Y и оси V в одном осевом направлении.
...	
N200 G1 Y22 V25 F200	; Наложение зависимого и независимого движения буксируемой оси V.
...	
TRAILOF(V,Y)	; Отключение 1-й структуры буксировок.
TRAILOF(W,Z)	; Отключение 2-й структуры буксировок.

Дополнительная информация

Типы осей

Структура буксировок может состоять из любых комбинаций линейных и круговых осей. В качестве ведущей оси при этом может быть определена и симулированная ось.

Буксируемые оси

С одной буксируемой осью одновременно может быть согласовано макс. 2 ведущие оси. Согласование осуществляется в различных структурах буксировок.

Буксируемая ось может быть запрограммирована со всеми доступными командами движения (G0, G1, G2, G3, ...). В дополнение к независимо определенным ходам буксируемая ось перемещается по ходам, полученным с помощью коэффициента связи из ее ведущих осей.

Динамическое ограничение

Динамическое ограничение зависит от типа активации структуры буксировок:

- **Активация в программе обработки детали**
Если активация выполняется в программе обработки детали и все ведущие оси находятся в активирующем канале как программные оси, то при перемещении ведущих осей динамика буксируемых осей учитывается таким образом, что не возникает перегрузок буксируемых осей.
Если активация выполняется в программе обработки детали с ведущими осями, которые не активны в активирующем канале как программные оси (\$AA_TYP ≠ 1), то при перемещении ведущих осей динамика буксируемой оси не учитывается. Из-за этого у буксируемых осей с меньшей, чем требуемая для связи, динамикой, возможна перегрузка.
- **Активация в синхронном действии**
Если активация выполняется в синхронном действии, то при перемещении ведущих осей динамика буксируемых осей не учитывается. Из-за этого у буксируемых осей с меньшей, чем требуемая для связи, динамикой, возможна перегрузка.

⚠ ОСТОРОЖНО**Перегрузка оси**

Если структура буксировок активируется

- в синхронных действиях
- в программе обработки детали с ведущими осями, которые не являются программными осями в канале буксируемой оси,

то пользователь/изготовитель станка должен предусмотреть подходящие меры по недопущению перегрузки буксируемых осей из-за движений перемещения ведущей оси.

Состояние связи

Состояние связи оси может быть запрошено в программе обработки детали с помощью системной переменной:

\$AA_COUP_ACT[<ось>]

Величина	Значение
0	нет активного соединения
8	буксировка активна

Индикация остаточного пути буксируемой оси для круговых осей модуло

Если ведущая и буксируемая ось это круговые оси модуло, то движения перемещения в ведущей оси суммируются $n \cdot 360^\circ$ где $n = 1, 2, 3...$, на индикации остаточного пути буксируемой оси для отключения соединения.

Пример: Сегмент программы с TRAILON и ведущей осью В и ведомой осью С

Программный код	Комментарий
TRAILON (C, B, 1)	; включить соединение
G0 B0	; Исходное положение

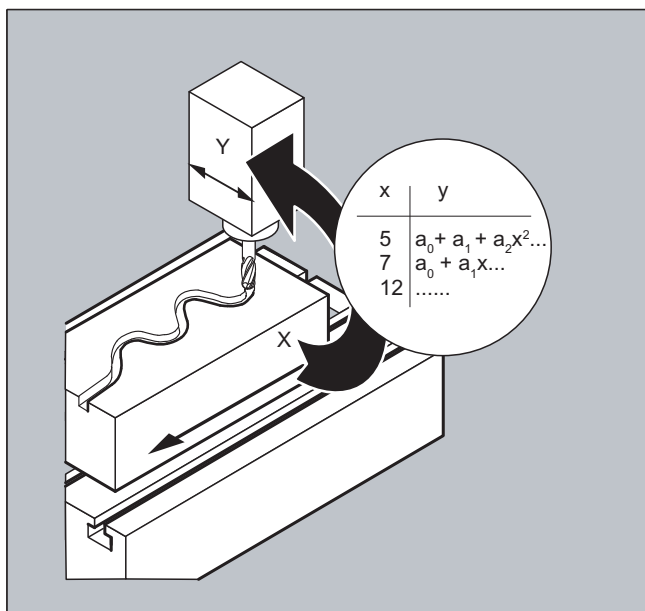
Программный код	Комментарий
	; Индикация остаточного пути на начале кадра:
G91 B360	; B=360, C=360
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

13.2 Таблицы кривых (СТАВ)

С помощью таблиц кривых могут быть запрограммированы позиционные и скоростные отношения между двумя осями (ведущая и ведомая ось). Определение таблиц кривых осуществляется в программе обработки детали.

Использование

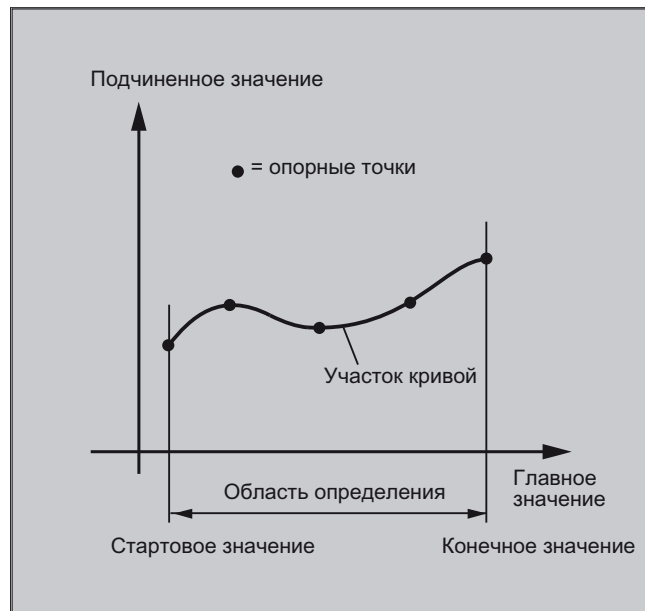
Таблицы кривых заменяют механические дисковые кулачки. Таблица кривых при этом создает основу для осевого соединения по главному значению, оформляя функциональную связь между главным и подчиненным значением: СЧПУ при соответствующем программировании вычисляет из согласованных друг с другом позиций ведущей и ведомой оси полином, соответствующий дисковому кулачку.



13.2.1 Определение таблиц кривых (СТАBDEF, CATBEND)

Таблица кривых представляет собой программу обработки детали или сегмент программы обработки детали, обозначенный препозицией СТАBDEF и завершающей командой CATBEND.

Внутри этого сегмента программы обработки детали через операторы движения отдельным позициям ведущей оси присваиваются однозначные позиции ведомой оси, служащие в качестве опорных точек для вычисления линии кривой в форме полинома макс. 5-го порядка.



Условие

Для определения таблиц кривых через соответствующее конфигурирование MD должно быть зарезервировано место в памяти (→ Изготовитель станка!).

Синтаксис

```
STABDEF (<ведомая ось>, <ведущая ось>, <n>, <периодичность>[, <место сохранения>])
...
STABEND
```

Значение

STABDEF ():	Начало определения таблиц кривых	
STABEND:	Конец определения таблиц кривых	
<ведомая ось>:	Ось, движение которой должно быть рассчитано через таблицу кривых	
<ведущая ось>:	Ось, предоставляющая главные значения для расчета движения ведомой оси	
<n>:	Номер (ID) таблицы кривых Номер таблицы кривых является однозначным и не зависит от места сохранения. Таблицы с одинаковым номером не могут находиться в статической и динамической памяти.	
<периодичность>:	Периодичность таблицы	
	0	Таблица не периодическая (обрабатывается только один раз, и для круговых осей)
	1	Таблица периодическая относительно ведущей оси
	2	Таблица периодическая относительно ведущей оси и ведомой оси

13.2 Таблицы кривых (CTAB)

<место сохранения>:	Указание места сохранения (опция)	
	"SRAM"	Таблица кривых создается в статической памяти ЧПУ.
	"DRAM"	Таблица кривых создается в динамической памяти ЧПУ.
Примечание: Если значение для этого параметра не программируется, то используется установленное с MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE стандартное место сохранения.		

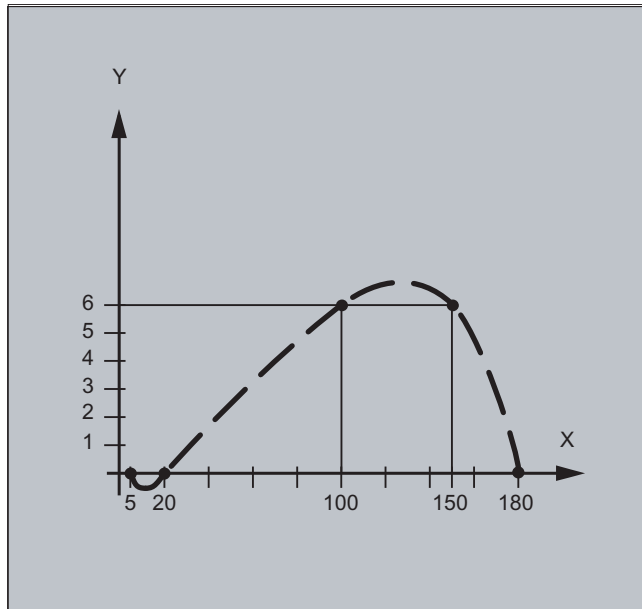
Примечание**Перезапись**

Таблица кривых перезаписывается, как только при повторном определении таблиц используется ее номер (<n>) (исключение: Таблица кривых активна в соединении осей или заблокирована с CTABLOCK). **При замене соответствующее предупреждение не выводится!**

Примеры**Пример 1: Сегмент программы как определение таблиц кривых**

Сегмент программы должен быть использован без изменений для определения таблицы кривых. Встречающаяся там команда для остановки предварительной обработки STOPRE может быть остановлена и снова сразу же активируется, как только сегмент программы более не будет использоваться для определения таблиц и CTABDEF и CTABEND были удалены

Программный код	Комментарий
...	
CTABDEF (Y, X, 1, 1)	; Определение таблицы кривых.
...	
IF NOT (\$P_CTABDEF)	
stopre	
ENDIF	
...	
CTABEND	

Пример 2: Определение не периодической таблицы кривых

Программный код	Комментарий
N100 STABDEF (Y,X,3,0)	; Начало определения не периодической таблицы кривых с номером 3.
N110 X0 Y0	; 1.Оператор движения, определяет стартовые значения и 1-ю опорную точку: главное значение: 0, подчиненное значение: 0
N120 X20 Y0	; 2-я опорная точка: главное значение: 0...20, подчиненное значение: Стартовое значение...0
N130 X100 Y6	; 3-я опорная точка: главное значение: 20...100, подчиненное значение: 0...6
N140 X150 Y6	; 4-я опорная точка: главное значение: 100...150, подчиненное значение: 6...6
N150 X180 Y0	; 5-я опорная точка: главное значение: 150...180, подчиненное значение: 6...0
N200 STABEND	; Конец определения. Таблица кривых в своем внутреннем представлении создается как полином макс. 5-го порядка. Расчет участка кривой с указанными опорными точками зависит от выбранного модального типа интерполяции (круговая, линейная, сплайн-интерполяция). Состояние программы обработки детали перед началом определения восстанавливается.

Пример 3: Определение периодической таблицы кривых

Определение периодической таблицы кривых с номером 2, диапазон главного значения от 0 до 360, движение ведомой оси от 0 до 45 градусов и обратно на 0:

Программный код	Комментарий
N10 DEF REAL DEPOS	
N20 DEF REAL GRADIENT	

13.2 Таблицы кривых (CTAB)

Программный код	Комментарий
N30 CTABDEF (Y, X, 2, 1)	; Начало определения.
N40 G1 X=0 Y=0	
N50 POLY	
N60 PO[X]=(45.0)	
N70 PO[X]=(90.0) PO[Y]=(45.0, 135.0, -90)	
N80 PO[X]=(270.0)	
N90 PO[X]=(315.0) PO[Y]=(0.0, -135.0, 90)	
N100 PO[X]=(360.0)	
N110 CTABEND	; Конец определения.
;Тест кривой через связь Y с X:	
N120 G1 F1000 X0	
N130 LEADON (Y, X, 2)	
N140 X360	
N150 X0	
N160 LEADOF (Y, X)	
N170 DEPPOS=CTAB (75.0, 2, GRADIENT)	; Чтение табличной функции для главного значения 75.0
N180 G0 X75 Y=DEPPOS	; Позиционирование ведущей и ведомой оси.
;После включения связи синхронизация ведомой оси не требуется.	
N190 LEADON (Y, X, 2)	
N200 G1 X110 F1000	
N210 LEADOF (Y, X)	
N220 M30	

Дополнительная информация

Стартовое и конечное значение таблицы кривых

Стартовым значением для начала области определения таблицы кривых считается первое указание связанных позиций осей (первый оператор движения) внутри определения таблицы кривых. Конечное значение области определения таблицы кривых определяется через последнюю команду перемещения соответственно.

Доступная мощность языка

Внутри определения таблиц кривых доступна вся мощность языка ЧПУ.

Примечание

Следующие данные недопустимы в определениях таблиц кривых:

- Остановка предварительной обработки
- Скачки в движении ведущих осей (к примеру, при смене трансформаций)
- Оператор движения только для ведомой оси
- Реверсирование ведущей оси, т. е. позиция ведущей оси всегда должна быть однозначной
- Операторы STABDEF и STABEND на различных программных уровнях.

Активность модальных операторов

Все действующие модально операторы, встречающиеся внутри определения таблицы кривых, по завершении определения таблицы становятся недействительными. Поэтому программа обработки детали, в которой выполняется определение таблицы, до и после определения таблицы находится в идентичном состоянии.

Присваивания R-параметрам

Присваивания R-параметрам внутри определения таблиц после STABEND отменяются.

Пример:

Программный код	Комментарий
...	
R10=5 R11=20	; R10=5
...	
STABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	; R10=25
X=R10	
STABEND	
...	; R10=5

Активация ASPLINE, BSPLINE, CSPLINE

Если внутри определения таблицы кривых STABDEF ... STABEND активируется ASPLINE, BSPLINE или CSPLINE, то перед этой активацией сплайна нужно запрограммировать как минимум одну стартовую точку. Немедленной активации после STABDEF следует избегать, поскольку иначе сплайн зависит от актуальной позиции оси перед определением таблицы кривых.

Пример:

Программный код
...
STABDEF (Y,X,1,0)

Программный код

```

X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND

```

Повторное использование таблиц кривых

Вычисленная через таблицу кривых функциональная связь ведущей и ведомой оси сохраняется под выбранным номером таблицы и после завершения программы обработки детали и после POWER OFF, если таблица находится в статической памяти ЧПУ (SRAM).

Таблица, созданная в динамической памяти (DRAM), удаляется при POWER OFF и при необходимости должна быть создана заново.

Созданная однажды таблица кривых может применяться к любым комбинациям ведущей и ведомой оси и не зависит от того, какие оси используются для создания таблицы кривых.

Замена таблиц кривых

Таблица кривых заменяется, как только при повторном определении таблиц используется ее номер.

Исключение: Таблица кривых активна в соединении осей или заблокирована с CTABLOCK.

Примечание

При замене таблиц кривых соответствующее предупреждение не выводится!

Определение таблиц кривых активно?

С помощью системной переменной \$P_CTABDEF из программы обработки детали можно запросить, активно ли определение таблиц кривых.

Отмена определения таблиц кривых

Сегмент программы обработки детали после вычленения операторов для определения таблиц кривых снова может использоваться как реальная программа обработки детали.

Загрузка таблиц кривых через "Выполнение с внешнего устройства"

При внешней обработке таблиц кривых размер буфера догрузки (DRAM) должен быть выбран через MD18360 \$MN_MM_EXT_PROG_BUFFER_SIZE таким образом, чтобы все определение таблиц кривых одновременно могло бы быть помещено в буфер догрузки. В ином случае выполнение программы обработки детали отменяется с ошибкой.

Скачки ведомой оси

В зависимости от установки машинных данных:
MD20900 \$MC_CTAB_ENABLE_NO_LEADMOTION

можно установить допуск для скачков ведомой оси при отсутствующем движении ведущей оси.

13.2.2 Проверка наличия таблицы кривых (STABEXISTS)

С помощью команды STABEXISTS можно проверить, имеется ли определенный номер таблицы кривых в памяти ЧПУ.

Синтаксис

STABEXISTS (<n>)

Значение

STABEXISTS:	Проверяет, имеется ли таблица кривых с номером <n> в статической или динамической памяти ЧПУ	
	0	Таблица не существует
	1	Таблица существует
<n>:	Номер (ID) таблицы кривых	

13.2.3 Удаление таблиц кривых (STABDEL)

С помощью STABDEL можно удалять таблицы кривых.

Примечание

Таблиц кривых, активные в соединении осей, не могут быть удалены.

Синтаксис

STABDEL (<n>)
 STABDEL (<n>, <m>)
 STABDEL (<n>, <m>, <место сохранения>)
 STABDEL ()
 STABDEL (, , <место сохранения>)

Значение

STABDEL:	Команда удаления таблиц кривых
<n>:	Номер (ID) удаляемой таблицы кривых При удалении области таблиц кривых STABDEL (<n>, <m>) с <n> указывается номер первой таблицы кривых области.
<m>:	При удалении области таблиц кривых STABDEL (<n>, <m>) с <m> указывается номер последней таблицы кривых области. <m> должна быть больше <n>!

13.2 Таблицы кривых (СТАВ)

<место сохранения>:	Указание места сохранения (опция)	
	При удалении без указания места сохранения, указанные таблицы кривых удаляются в статической и динамической памяти ЧПУ.	
	При удалении с указанием места сохранения, из указанных таблиц кривых удаляются только те, которые находятся в указанной памяти. Остальные сохраняются.	
	"SRAM"	Удаление в статической памяти ЧПУ
	"DRAM"	Удаление в динамической памяти ЧПУ

Если СТАBDEL программируется без указания удаляемой таблицы кривых, то удаляются **все** таблицы кривых или все таблицы кривых в указанной памяти:

СТАBDEL () :	Удаляет все таблицы кривых в статической и динамической памяти ЧПУ
СТАBDEL (, , "SRAM") :	Удаляет все таблицы кривых в статической памяти ЧПУ
СТАBDEL (, , "DRAM") :	Удаляет все таблицы кривых в динамической памяти ЧПУ

Примечание

Если при многократном удалении СТАBDEL (<n>, <m>) или СТАBDEL () минимум одна из удаляемых таблиц кривых активна в соединении, то команда удаления не выполняется, т. е. **ни одна** из адресованных таблиц кривых не удаляется.

13.2.4 Блокировка таблиц кривых от удаления и перезаписи (СТАBLOCK, СТАBUNLOCK)

Таблицы кривых через установку блокировок могут быть защищены от непреднамеренного удаления и перезаписи. Установленная блокировка снова может быть снята в любое время.

Синтаксис**Установить блокировку:**

```
СТАBLOCK (<n>)
СТАBLOCK (<n>, <m>)
СТАBLOCK (<n>, <m>, <место сохранения>)
СТАBLOCK ( )
СТАBLOCK ( , , <место сохранения>)
```

Снять блокировку:

```
СТАBUNLOCK (<n>)
СТАBUNLOCK (<n>, <m>)
СТАBUNLOCK (<n>, <m>, <место сохранения>)
СТАBUNLOCK ( )
СТАBUNLOCK ( , , <место сохранения>)
```

Значение

STABLOCK:	Команда для установки блокировки от удаления/перезаписи
STABUNLOCK:	Команда для снятия блокировки от удаления/перезаписи STABUNLOCK снова разрешает заблокированные с STABLOCK таблицы кривых. Таблицы кривых, действующие в активном соединении, остаются заблокированными и не могут быть удалены. Блокировка с STABLOCK отменена, как только блокирование через активное соединение отменяется с деактивацией соединения. Таким образом, эта таблица может быть удалена. Повторного вызова STABUNLOCK не требуется.
<n>:	Номер (ID) блокируемой/разблокируемой таблицы кривых При блокировке/разблокировке области таблиц кривых STABLOCK (<n>, <m>) / STABUNLOCK (<n>, <m>) с <n> указывается номер первой таблицы кривых области.
<m>:	При блокировке/разблокировке области таблиц кривых STABLOCK (<n>, <m>) / STABUNLOCK (<n>, <m>) с <m> указывается номер последней таблицы кривых области. <m> должна быть больше <n>!
<место сохранения>:	Указание места сохранения (опция) При установке/снятии блокировки без указания места сохранения, указанные таблицы кривых блокируются/разблокируются в статической и динамической памяти ЧПУ. При установке/снятии блокировки с указанием места сохранения, из указанных таблиц кривых блокируются/разблокируются только те, которые находятся в указанной памяти. Прочие не блокируются/разблокируются.
	"SRAM" Установить/снять блокировку в статической памяти ЧПУ
	"DRAM" Установить/снять блокировку в динамической памяти ЧПУ

Если STABLOCK/STABUNLOCK программируется без указания блокируемой/разблокируемой таблицы кривых, то блокируются/разблокируются **все** таблицы кривых или все таблицы кривых в указанной памяти:

STABLOCK () :	Блокирует все таблицы кривых в статической и динамической памяти ЧПУ
STABLOCK (, , "SRAM") :	Блокирует все таблицы кривых в статической памяти ЧПУ
STABLOCK (, , "DRAM") :	Блокирует все таблицы кривых в динамической памяти ЧПУ
STABUNLOCK () :	Деблокирует все таблицы кривых в статической и динамической памяти ЧПУ
STABUNLOCK (, , "SRAM") :	Деблокирует все таблицы кривых в статической памяти ЧПУ
STABUNLOCK (, , "DRAM") :	Деблокирует все таблицы кривых в динамической памяти ЧПУ

13.2.5 Таблицы кривых: Определение свойств таблиц (STABID, STABISLOCK, STABMEMTYP, STABPERIOD)

С помощью этих команд можно запросить важные свойства таблицы кривых (номер таблицы, состояние блокировки, место сохранения, периодичность).

13.2 Таблицы кривых (СТАВ)

Синтаксис

СТАВИD (<p>)
 СТАВИD (<p>, <место сохранения>)
 СТАВИSLOCK (<n>)
 СТАВИMEMTYP (<n>)
 ТАВPERIOD (<n>)

Значение

СТАВИD:	Выводит номер таблицы , внесенной в указанную память в качестве <p>-ной таблицы кривых. Пример: СТАВИD (1, "SRAM") возвращает номер первой таблицы кривых в статической памяти ЧПУ. Первая таблица кривых при этом соответствует таблице кривых со старшим номером таблицы. Примечание: Если между следующими друг за другом вызовами СТАВИD последовательность таблиц кривых в памяти изменяется, к примеру, из-за удаления таблиц кривых с СТАВДЕL, то СТАВИD (<p>, . . .) может вернуть с тем же номером <p> отличную от прежней таблицу кривых.	
СТАВИSLOCK:	Возвращает состояние блокировки таблицы кривых с номером <n>:	
	0	Таблица не заблокирована
	1	Таблица заблокирована через СТАВLOCK
	2	Таблица заблокирована через активное соединение
	3	Таблица заблокирована через СТАВLOCK и активное соединение
	-1	Таблица не существует
СТАВИMEMTYP:	Возвращает место сохранения таблицы кривых с номером <n>:	
	0	Таблица в статической памяти ЧПУ
	1	Таблица в динамической памяти ЧПУ
	-1	Таблица не существует
СТАВPERIOD:	Возвращает периодичность таблицы кривых с номером <n>:	
	0	Таблица не периодическая
	1	Таблица периодическая в ведущей оси
	2	Таблица периодическая в ведущей и ведомой оси
	-1	Таблица не существует
<p>:	Номер элемента в памяти	
<n>:	Номер (ID) таблицы кривых	
<место сохранения>:	Указание места сохранения (опция)	
	"SRAM"	Статическая память ЧПУ
	"DRAM"	Динамическая память ЧПУ
	Примечание: Если значение для этого параметра не программируется, то используется установленное с MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE стандартное место сохранения.	

13.2.6 Чтение значений таблиц кривых (СТАБTSV, СТАБTEV, СТАБTSP, СТАБТЕР, СТАБSSV, СТАБSEV, СТАВ, СТАБINV, СТАБTMIN, СТАБTMAX)

Следующие значения таблиц кривых могут считываться в программе обработки детали:

- Значения ведомой и ведущей оси в начале и конце таблицы кривых
- Значения ведомой оси в начале и конце сегмента кривой
- Значение ведомой оси к значению ведущей оси
- Значение ведущей оси к значению ведомой оси
- Мин. и макс. значение ведомой оси
 - во всей области определений таблицы кривых или
 - на определенном интервале таблицы кривых

Синтаксис

```
СТАБTSV (<n>, <градиент>[, <ведомая ось>])
СТАБTEV (<n>, <градиент>[, <ведомая ось>])
СТАБTSP (<n>, <градиент>[, <ведущая ось>])
СТАБТЕР (<n>, <градиент>[, <ведущая ось>])
СТАБSSV (<главное значение>, <n>, <градиент>[, <ведомая ось>])
СТАБSEV (<главное значение>, <n>, <градиент>[, <ведомая ось>])
СТАВ (<главное значение>, <n>, <градиент>[, <ведомая ось>, <ведущая ось>])
СТАБINV (<подчиненное значение>, <приближенное
значение>, <n>, <градиент>[, <ведомая ось>, <ведущая ось>])
СТАБTMIN (<n> [, <ведомая ось>])
СТАБTMAX (<n> [, <ведомая ось>])
СТАБTMIN (<n>, <a>, <b> [, <ведомая ось>, <ведущая ось>])
СТАБTMAX (<n>, <a>, <b> [, <ведомая ось>, <ведущая ось>])
```

Значение

СТАБTSV:	Читать значение ведомой оси в начале таблицы кривых Nr. <n>
СТАБTEV:	Читать значение ведомой оси в конце таблицы кривых Nr. <n>
СТАБTSP:	Читать значение ведущей оси в начале таблицы кривых Nr. <n>
СТАБТЕР:	Читать значение ведущей оси в конце таблицы кривых Nr. <n>
СТАБSSV:	Читать значение ведомой оси в начале относящегося к указанному значению ведущей оси (<главное значение>) сегмента кривой
СТАБSEV:	Читать значение ведомой оси в конце относящегося к указанному значению ведущей оси (<главное значение>) сегмента кривой
СТАВ:	Читать значение ведомой оси к указанному значению ведущей оси (<главное значение>)
СТАБINV:	Читать значение ведущей оси к указанному значению ведомой оси (<подчиненное значение>)

13.2 Таблицы кривых (СТАВ)

СТАВТMIN:	Определить мин. значение ведомой оси: - во всей области определений таблицы кривых или - на определенном интервале <a> ...
СТАВТMAX:	Определить макс. значение ведомой оси: - во всей области определений таблицы кривых или - на определенном интервале <a> ...
<n>:	Номер (ID) таблицы кривых
<градиент>:	В параметре <градиент> возвращается градиент функции таблицы кривых на полученной позиции
<ведомая ось>:	Ось, движение которой должно быть рассчитано через таблицу кривых (опция)
<ведущая ось>:	Ось, предоставляющая главные значения для расчета движения ведомой оси (опция)
<подчиненное значение>:	Значение ведомой оси для чтения соответствующего значения ведущей оси при СТАВINV
<главное значение>:	Значение ведущей оси: - для чтения соответствующего значения ведомой оси при СТАВ или - для выбора сегмента кривой при СТАВSSV/СТАВSEV
<приближенное значение>:	Согласование значения ведущей оси со значением ведомой оси при СТАВINV не всегда должно быть однозначным. Поэтому для СТАВINV в качестве параметра необходимо приближенное значение для ожидаемого значения ведущей оси.
<a>:	Нижняя граница интервала главного значения при СТАВТMIN/СТАВТMAX
:	Верхняя граница интервала главного значения при СТАВТMIN/СТАВТMAX Примечание: Интервал главного значения <a> ... не должен выходить за область определения таблицы кривых.

Примеры

Пример 1:

Определить значения ведомой и ведущей оси в начале и конце таблицы кривых, а также мин. и макс. значение ведомой оси во всей области определения таблицы кривых.

Программный код	Комментарий
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL STARTPARA	
N40 DEF REAL ENDPARA	
N50 DEF REAL MINVAL	
N60 DEF REAL MAXVAL	
N70 DEF REAL GRADIENT	

Программный код	Комментарий
...	
N100 STABDEF (Y,X,1,0)	; начало определения таблицы
N110 X0 Y10	; стартовая позиция, 1-й сегмент таблицы
N120 X30 Y40	; конечная позиция 1-го сегмента таблицы = стартовая позиция 2-ого сегмента таблицы
N130 X60 Y5	; конечная позиция 2-й сегмент таблицы = ...
N140 X70 Y30	
N150 X80 Y20	
N160 STABEND	; Конец определения таблицы.
...	
N200 STARTPOS=CTABTSV (1, GRADIENT)	; Значение ведомой оси на начале таблицы кривых = 10
N210 ENDPOS=CTABTEV (1, GRADIENT)	; Значение ведомой оси на конце таблицы кривых = 20
N220 STARTPARA=CTABTSP (1, GRADIENT)	; Значение ведущей оси на начале таблицы кривых = 0
N230 ENDPARA=CTABTEP (1, GRADIENT)	; Значение ведущей оси на конце таблицы кривых = 80
N240 MINVAL=CTABTMIN (1)	; Мин. значение ведомой оси при Y=5
N250 MAXVAL=CTABTMAX (1)	; Макс. значение ведомой оси при Y=40

Пример 2:

Определение значений ведомой оси в начале и конце относящегося к значению ведущей оси X=30 сегмента кривой.

Программный код	Комментарий
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
...	
N100 STABDEF (Y,X,1,0)	; Начало определения таблицы.
N110 X0 Y0	; стартовая позиция, 1-й сегмент таблицы
N120 X20 Y10	; конечная позиция 1-го сегмента таблицы = стартовая позиция 2-ого сегмента таблицы
N130 X40 Y40	; конечная позиция 2-й сегмент таблицы = ...
N140 X60 Y10	
N150 X80 Y0	
N160 STABEND	; Конец определения таблицы.
...	
N200 STARTPOS=CTABSSV (30.0,1, GRADIENT)	; стартовая позиция Y во 2-м сегменте = 10
N210 ENDPOS=CTABSEV (30.0,1, GRADIENT)	; конечная позиция Y во 2-м сегменте = 40

Дополнительная информация**Использование в синхронных действиях**

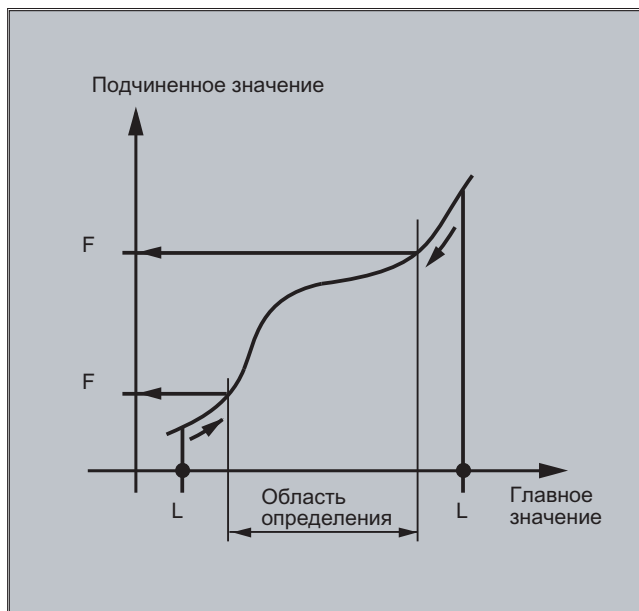
Все команды по чтению значений таблиц кривых могут использоваться и в синхронных действиях (см. также главу "Синхронные действия движения").

При использовании команд СТАВИНВ, СТАВТМІН и СТАВТМАХ следить за тем, чтобы:

- на момент выполнения было бы достаточно мощности ЧПУ или
- перед вызовом запросить число сегментов таблицы кривых, чтобы при необходимости можно было бы разделить данную таблицу.

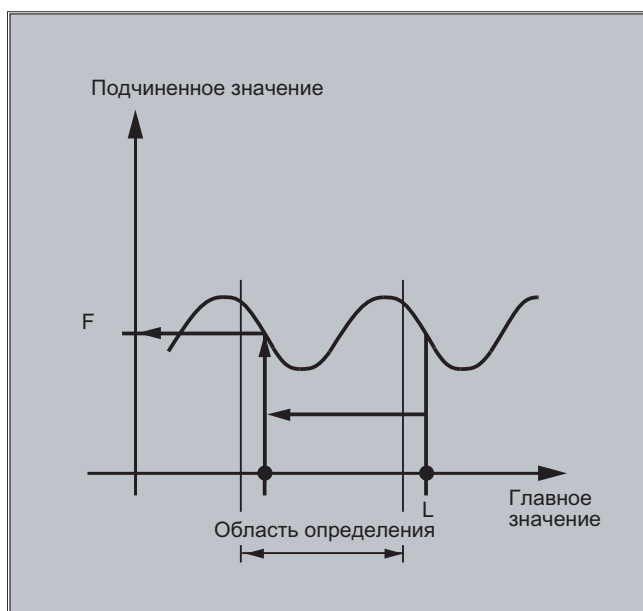
СТАВ для не периодических таблиц кривых

Если указанное <главное значение> лежит за пределами области определения, то в качестве подчиненного значения выводится верхняя или нижняя граница:



СТАВ для периодических таблиц кривых

Если <главное значение> лежит вне области определения, то главное значение модулю области определения нормируется и соответствующее подчиненное значение выводится.



Приближенное значение для STABINV

Для команды STABINV необходимо приближенное значение для ожидаемого главного значения. STABINV возвращает главное значение, наиболее близкое к приближенному значению. Приближенным значением может быть, к примеру, главное значение из предшествующего такта интерполятора.

Градиент функции таблицы кривых

Вывод градиента (<градиент>) позволяет рассчитать скорость ведущей или ведомой оси на соответствующей позиции.

Указание ведущей или ведомой оси

Опциональное указание ведущей и/или ведомой оси важно, т.к. ведущая и ведомая оси сконфигурированы в различных единицах длины.

STABSSV, STABSEV

Команды STABSSV и STABSEV в следующих случаях **не** подходят для опроса запрограммированных сегментов:

- Запрограммированы окружности или эвольвенты.
- Фаска или закругление активны с CHF/RND.
- Перешлифовка с G643 активна.
- Компрессор кадров ЧПУ, к примеру, с COMPON/COMPCURV/COMPCAD активен.

13.2.7 Таблицы кривых: Проверка использования ресурсов (STABNO, STABNOMEM, STABFNO, STABSEGID, STABSEG, STABFSEG, STABMSEG, STABPOLID, STABPOL, STABFPOL, STABMPOL)

С помощью этих команд программист может получить актуальную информацию об использовании ресурсов для таблиц кривых, сегментов таблиц и полиномов.

Синтаксис

```

STABNO
STABNOMEM (<место сохранения>)
STABFNO (<место сохранения>)
STABSEGID (<n>, <место сохранения>)
STABSEG (<место сохранения>, <тип сегмента>)
STABFSEG (<место сохранения>, <тип сегмента>)
STABMSEG (<место сохранения>, <тип сегмента>)
STABPOLID (<n>)
STABPOL (<место сохранения>)
STABFPOL (<место сохранения>)
STABMPOL (<место сохранения>)

```

Значение

STABNO:	Определить общее число определенных таблиц кривых (в статической и динамической памяти ЧПУ)
STABNOMEM:	Определить число определенных таблиц кривых в указанном <месте сохранения>
STABFNO:	Определить число еще возможных таблиц кривых в указанном <месте сохранения>
STABSEGID:	Определить число сегментов кривой указанного <типа сегмента>, которые используются таблицей кривых с номером <n>
STABSEG:	Определить число использованных сегментов кривой указанного <типа сегмента> в указанном <месте сохранения>
STABFSEG:	Определить число еще возможных сегментов кривой указанного <типа сегмента> в указанном <месте сохранения>
STABMSEG:	Определить число макс. возможных сегментов кривой указанного <типа сегмента> в указанном <месте сохранения>
STABPOLID:	Определить число криволинейных полиномов, которые используются таблицей кривых с номером <n>
STABPOL:	Определить число использованных криволинейных полиномов в указанном <месте сохранения>
STABFPOL:	Определить число еще возможных криволинейных полиномов в указанном <месте сохранения>
STABMPOL:	Определить число макс. возможных криволинейных полиномов в указанном <месте сохранения>
<n>:	Номер (ID) таблицы кривых

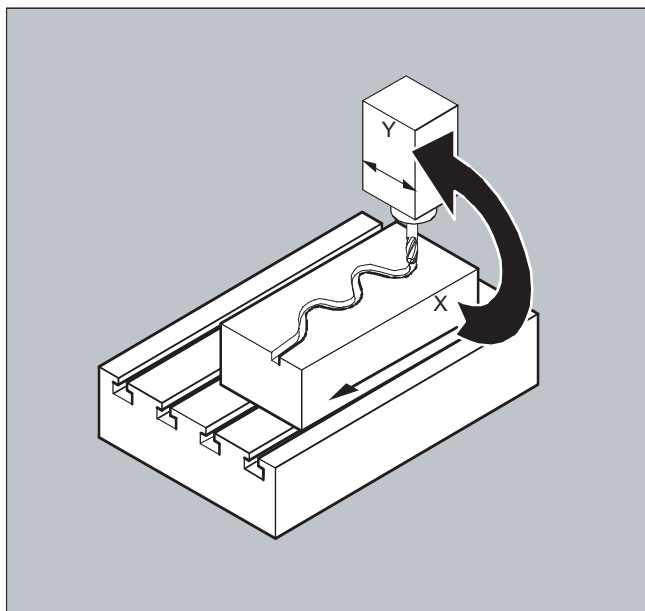
<место сохранения>:	Указание места сохранения (опция)	
	"SRAM"	Статическая память ЧПУ
	"DRAM"	Динамическая память ЧПУ
	Примечание: Если значение для этого параметра не программируется, то используется установленное с MD20905 \$MC_STAB_DEFAULT_MEMORY_TYPE стандартное место сохранения.	
<тип сегмента>:	Указание типа сегмента (опция)	
	"L"	Линейные сегменты
	"P"	Полиномиальные сегменты
	Примечание: Если значение для этого параметра не программируется, то выводится сумма из линейных и полиномиальных сегментов.	

13.3 Осевое соединение по главному значению (LEADON, LEADOF)

Примечание

Эта функция недоступна для SINUMERIK 828D!

При осевом соединении по главному значению одна ведущая и одна ведомая ось перемещаются синхронно. При этом соответствующая позиция ведомой оси через таблицу кривых или вычисленный из нее полином однозначно согласована с – возможно симулированной – позицией ведущей оси.



Ведущей осью называется та ось, которая выводит входные значения для таблицы кривых. **Ведомой осью** называется ось, занимающая вычисленные через таблицу кривых позиции.

Соединение по фактическому и заданному значению

В качестве главных значений, т. е. выходных значений для определения позиции ведомой оси, могут использоваться:

- Фактические значения позиции ведущей оси: соединение по фактическому значению
- Заданные значения позиции ведущей оси: соединение по заданному значению

Соединение по главному значению всегда действует в базовой кинематической системе.

По созданию таблиц кривых см. главу "Таблицы кривых".

Синтаксис

LEADON (<ведомая ось>,<ведущая ось>,<n>)

LEADOF (<ведомая ось>,<ведущая ось>)

или отключение без указания ведущей оси:

LEADOF (<ведомая ось>)

13.3 Осевое соединение по главному значению (LEADON, LEADOF)

Включение и выключение соединения по главному значению возможно как из программы обработки детали, так и при движении из синхронных действий.

Значение

LEADON:	Включить соединение по главному значению
LEADOF:	Выключить соединение по главному значению
<ведомая ось>:	Ведомая ось
<ведущая ось>:	Ведущая ось
<n>:	Номер таблиц кривых
\$SA_LEAD_TYPE:	Переключение между соединением по заданному и фактическому значению

Отключение соединения по главному значению, LEADOF

При отключении соединения по главному значению ведомая ось снова становится обычной командной осью!

Осевое соединение по главному значению и различные рабочие состояния, RESET

В зависимости от установки в машинных данных, соединения по главному значению при RESET отключаются.

Пример соединения по главному значению из синхронного действия

На прессовальной установке обычное механическое соединение между ведущей осью (вал пуансона) и осями системы передачи из осей передачи и вспомогательных осей должно быть заменено на электронную соединительную систему.

Пример показывает, как на прессовальной установке механическая система передачи заменяется на электронную систему передачи. Процессы соединения и разъединения реализованы как **статические синхронные действия**.

От ведущей оси LW (вал пуансона) выполняется определенное управление осями передачи и вспомогательными осями как ведомыми осями через таблицы кривых.

Ведомые оси

X ось подачи или продольная ось
 YL замыкающая или поперечная ось
 ZL Hubachse
 U подача вальцов, вспомогательная ось
 V корректирующая головка, вспомогательная ось
 W смазка, вспомогательная ось

Действия

В качестве операций в синхронных действиях, к примеру, встречаются:

- Соединение, LEADON (<ведомая ось>,<ведущая ось>,<номер таблицы кривых>)
- Разъединение,LEADOF (<ведомая ось>,<ведущая ось>)
- Установка фактического значения, PRESETON (<ось>,<значение>)

13.3 Осевое соединение по главному значению (LEADON, LEADOF)

- Установка маркера, `$AC_MARKER[i]=<значение>`
- Тип соединения: реальное/виртуальное главное значение
- Подвод к позициям осей, `POS[<ось>]=<значение>`

Условия

В качестве условий обрабатываются цифровые быстрые входы, переменные реального времени `$AC_MARKER` и сравнения позиций, связанные логическим оператором И.

Примечание

В примере ниже переход строк, вставки и **жирный** шрифт используются только для того, чтобы улучшить читабельность программирования. Для СЧПУ все, стоящее под одним номером строки, является однострочным.

Комментарий

Программный код	Комментарий
	; Определяет все статические синхронные действия.
	; ****Сбросить маркер
N2 \$AC_MARKER[0]=0 \$AC_MARKER[1]=0 \$AC_MARKER[2]=0 \$AC_MARKER[3]=0 \$AC_MARKER[4]=0 \$AC_MARKER[5]=0 \$AC_MARKER[6]=0 \$AC_MARKER[7]=0	
	; **** E1 0=>1 соединение передача ВКЛ
N10 IDS=1 EVERY (\$A_IN[1]==1) AND (\$A_IN[16]==1) AND (\$AC_MARKER[0]==0) DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) \$AC_MARKER[0]=1	
	; **** E1 0=>1 соединение подача вальцов ВКЛ
N20 IDS=11 EVERY (\$A_IN[1]==1) AND (\$A_IN[5]==0) AND (\$AC_MARKER[5]==0) DO LEADON(U,LW,4) PRESETON(U,0) \$AC_MARKER[5]=1	
	; **** E1 0->1 соединение корректирующая головка ВКЛ
N21 IDS=12 EVERY (\$A_IN[1]==1) AND (\$A_IN[5]==0) AND (\$AC_MARKER[6]==0) DO LEADON(V,LW,4) PRESETON(V,0) \$AC_MARKER[6]=1	
	; **** E1 0->1 соединение смазка ВКЛ
N22 IDS=13 EVERY (\$A_IN[1]==1) AND (\$A_IN[5]==0) AND (\$AC_MARKER[7]==0) DO LEADON(W,LW,4) PRESETON(W,0) \$AC_MARKER[7]=1	
	; **** E2 0=>1 соединение ВЫКЛ
N30 IDS=3 EVERY (\$A_IN[2]==1) DO LEADOF(X,LW) LEADOF(YL,LW) LEADOF(ZL,LW) LEADOF(U,LW) LEADOF(V,LW) LEADOF(W,LW) \$AC_MARKER[0]=0 \$AC_MARKER[1]=0 \$AC_MARKER[3]=0 \$AC_MARKER[4]=0 \$AC_MARKER[5]=0 \$AC_MARKER[6]=0 \$AC_MARKER[7]=0 N110 G04 F01 N120 M30	

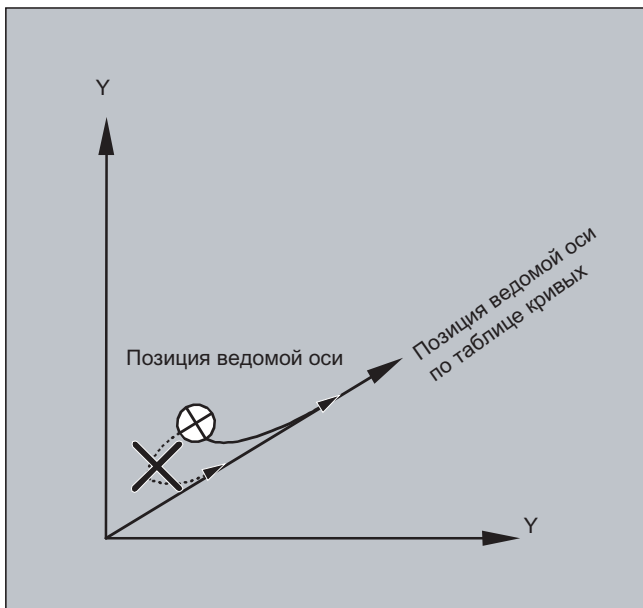
Описание

Для соединения по главному значению необходима синхронизация ведущей и ведомой оси. Эта синхронизация может быть достигнута только тогда, когда ведомая ось при включении соединения по главному значению находится в пределах диапазона допуска вычисленного из таблицы кривых участка кривой.

13.3 Осевое соединение по главному значению (LEADON, LEADOF)

Диапазон допуска для положения ведомой оси определен через машинные данные MD 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE .

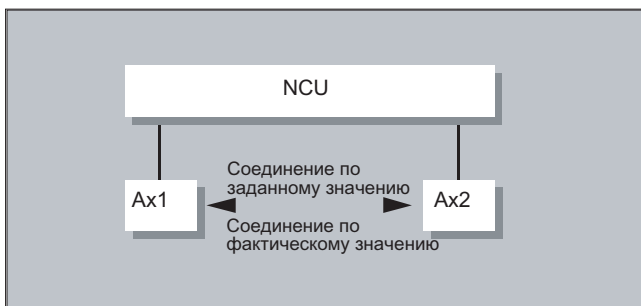
Если ведомая ось при включении соединения по главному значению еще не находится на соответствующей позиции, то синхронный ход восстанавливается автоматически, как только вычисленное заданное значение позиции для ведомой оси приближается к фактической позиции ведомой оси. При этом ведомая ось в процессе синхронизации перемещается в направлении, которое определено через заданную скорость ведомой оси (вычислена из скорости ведущей оси и по таблице кривых STAB).

**Нет синхронного хода**

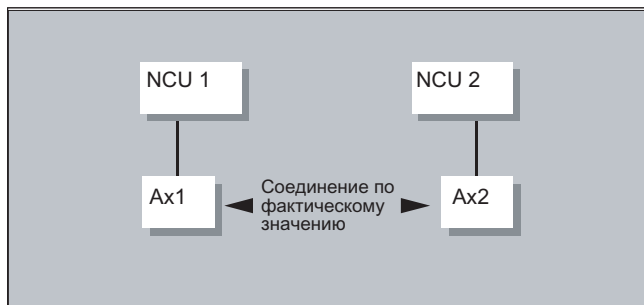
Если вычисленная заданная позиция ведомой оси при включении соединения по главному значению удаляется от актуальной позиции ведомой оси, то синхронный ход не устанавливается.

Соединение по фактическому и заданному значению

Соединение по заданному значению обеспечивает по сравнению с соединением по фактическому значению лучший синхронный ход между ведущей и ведомой осями и поэтому является стандартной предустановкой.



Соединение по заданному значению возможно только тогда, когда интерполяция ведущей и ведомой оси выполняется с одного NCU. В случае внешней ведущей оси ведомая ось может быть соединена с ведущей осью только через фактические значения.



Переключение возможно через установочные данные \$SA_LEAD_TYPE.

Переключение между соединением по фактическому и заданному значению всегда должно выполняться при остановленной ведомой оси. Т.к. только в состоянии покоя после переключения выполняется повторная синхронизация.

Пример использования

Безошибочное чтение фактических значений при сильных вибрациях станка невозможно. Поэтому при использовании соединения по главному значению в пресовом транспорте в рабочих операциях с сильными вибрациями может возникнуть необходимость переключения с соединения по фактическому значению на соединение по заданному значению.

Симуляция главного значения при соединении по заданному значению

Через машинные данные можно разделить интерполятор для ведущей оси и Servo. Таким образом, при соединении по заданному значению, заданные значения могут создаваться и без фактического движения ведущей оси.

Созданные через соединение по заданному значению главные значения для использования, к примеру, в синхронных действиях, могут считываться из следующих переменных:

- | | |
|---------------|----------------------------|
| - \$AA_LEAD_P | Главное значение, позиция |
| - \$AA_LEAD_V | Главное значение, скорость |

Создание главных значений

Главные значения по выбору могут создаваться посредством иных самостоятельно запрограммированных методов. Созданные таким образом главные значения записываются и считываются из переменных

- | | |
|----------------|----------------------------|
| - \$AA_LEAD_SP | Главное значение, позиция |
| - \$AA_LEAD_SV | Главное значение, скорость |

. Для использования этих переменных необходимы установочные данные \$SA_LEAD_TYPE = 2.

Состояние соединения

В программе ЧПУ состояние соединения может быть запрошено посредством следующей системной переменной:

\$AA_COUP_ACT[ось]

0: нет активного соединения

16: активно соединение по главному значению

Управление состоянием в синхронных действиях

Процессы переключения и соединения управляются через переменную реального времени:

\$AC_MARKER[i] = n

с помощью:

i номер меркера

n значение состояния

13.4 Электронный редуктор (EG)

С помощью функции "Электронный редуктор" можно управлять движением **ведомой оси** по линейному кадру движения в зависимости от макс. пяти **ведущих осей**. Связи между ведущими осями и ведомой осью определены через коэффициент связи для каждой ведущей оси.

Вычисленный компонент движения ведомой оси образуется посредством сложения из компонентов движения отдельных ведущих осей, умноженных на соответствующие коэффициенты связи. При активизации структуры осей электронного редуктора может быть запущена синхронизация ведомой оси на определенную позицию. Из программы обработки детали структура редуктора может:

- определяться,
- включаться,
- отключаться,
- удаляться

Движение ведомой оси может следовать по выбору из

- заданных значений ведущих осей, а также
- фактических значений ведущих осей.

В качестве расширения возможна реализация и нелинейных связей между ведущими осями и ведомой осью через **таблицы кривых** (см. главу "Параметры траектории"). Возможно каскадирование электронного редуктора, т. е. ведомая ось одного электронного редуктора может быть ведущей осью для следующего электронного редуктора.

13.4.1 Определение электронного редуктора (EGDEF)

Структура осей электронного редуктора определяется через указание ведомой оси и минимум одной и макс. пяти ведущих осей с соответствующим типом соединения.

Условие

Условие для определения структуры осей электронного редуктора:

Для ведомой оси еще не должно быть определено соединение осей (при необходимости сначала удалить существующее посредством EGDEL).

Синтаксис

EGDEF (ведомая ось,ведущая ось1,тип связи1,ведущая ось2,тип связи2,...)

Значение

EGDEF:	Определение электронного редуктора	
Ведомая ось:	Ось, управляемая ведущими осями	
Ведущая ось1 ,..., Ведущая ось5	Оси, управляющие ведомой осью	
Тип связи1 ,..., Тип связи5	Тип связи Тип связи не должен быть одинаковым для всех ведущих осей и поэтому указывается отдельно для каждой ведущей оси.	
	Величина:	Значение:
	0	Ведомая ось управляется фактическим значением соответствующей ведущей оси.
	1	Ведомая ось управляется заданным значением соответствующей ведущей оси.

Примечание

Коэффициенты связи при определении структуры соединения электронного редуктора предустанавливаются на ноль.

Примечание

EGDEF вызывает остановку предварительной обработки. Определение редуктора с EGDEF должно использоваться без изменений и тогда, когда в системах одна или несколько ведущих осей через **таблицу кривых** воздействуют на ведомую ось.

Пример

Программный код	Комментарий
EGDEF (C, B, 1, Z, 1, Y, 1)	; Определение структуры осей электронного редуктора. Ведущие оси B, Z, Y управляют ведомой осью C через заданное значение.

13.4.2 Включение электронного редуктора (EGON, EGONSYN, EGONSYNE)

Для включения структуры осей электронного редуктора существует 3 варианта.

Синтаксис

Вариант 1:

Включение структуры осей электронного редуктора по выбору без синхронизации с помощью:

EGON (FA, "Режим смены кадра", LA1, Z1, N1, LA2, Z2, N2, . . . , LA5, Z5, N5)

Вариант 2:

Включение структуры осей электронного редуктора по выбору с синхронизацией с помощью:

EGONSYN (FA, "Режим смены кадра", SynPosFA, [, LAi, SynPosLAI, Zi, Ni])

Вариант 3:

Включение структуры осей электронного редуктора по выбору с синхронизацией и задачей режима подвода с помощью:

EGONSYNE (FA, "Режим смены кадра", SynPosFA, режим подвода[, LAi, SynPosLAI, Zi, Ni])

Значение**Вариант 1:**

FA	Ведомая ось	
Режим смены кадра:	Возможно использование следующих режимов:	
	"NOC"	Смена кадра выполняется сразу же
	"FINE"	Смена кадра при "Синхронном ходе точном"
	"COARSE"	Смена кадра при "Синхронном ходе грубом"
	"IPOSTOP"	Смена кадра при синхронном ходе со стороны заданного значения
LA1, ... LA5	Ведущие оси	
Z1, ... Z5	Числитель для коэффициента связи i	
N1, ... N5	Знаменатель для коэффициента связи i Коэффициент связи i = числитель i/знаменатель i	

Могут быть запрограммированы только ведущие оси, которые до этого были специфицированы с EGDEF. Должна быть запрограммирована минимум одна ведущая ось.

Вариант 2:

FA	Ведомая ось	
Режим смены кадра:	Возможно использование следующих режимов:	
	"NOC"	Смена кадра выполняется сразу же
	"FINE"	Смена кадра при "Синхронном ходе точном"
	"COARSE"	Смена кадра при "Синхронном ходе грубом"
	"IPOSTOP"	Смена кадра при синхронном ходе со стороны заданного значения
[, LAi, SynPosLAI, Zi, Ni]	(не записывать квадратные скобки) Мин. 1, макс. 5 последовательностей:	
LA1, ... LA5	Ведущие оси	
SynPosLAI	Синхронная позиция для ведущей оси i.	
Z1, ... Z5	Числитель для коэффициента связи i	
N1, ... N5	Знаменатель для коэффициента связи i Коэффициент связи i = числитель i/знаменатель i	

Могут быть запрограммированы только ведущие оси, которые до этого были специфицированы с EGDEF. Через запрограммированные "синхронные позиции" для ведомой оси (SynPosFA) и для ведущих осей (SynPosLA), определяются позиции, в которых структура соединения считается *синхронной*. Если при включении электронный редуктор находится не в синхронном состоянии, то ведомая ось перемещается на свою определенную синхронную позицию.

Вариант 3:

Параметры соответствуют таковым варианта 2, плюс:

Режим подвода:	Возможно использование следующих режимов:	
	"NTGT"	Оптимизированный по времени подвод к следующей впадине между зубьями
	"NTGP"	Оптимизированный по ходу подвод к следующей впадине между зубьями
	"ACN"	Абсолютное перемещение круговой оси в отрицательном направлении вращения
	"ACP"	Абсолютное перемещение круговой оси в положительном направлении вращения
	"DCT"	С оптимизацией по времени к запрограммированной синхронной позиции
	"DCP"	С оптимизацией по ходу к запрограммированной синхронной позиции

Вариант 3 влияет только на ведомые оси модулю, связанные с ведущими осями модулю. Оптимизация по времени учитывает границы скорости ведомой оси.

Дополнительная информация

Описание вариантов включения

Вариант 1:

Позиции ведущих осей, а также ведомой оси, на момент включения сохраняются как "Синхронные позиции". "Синхронные позиции" могут быть считаны с помощью системной переменной \$AA_EG_SYN.

Вариант 2:

Если оси по модулю находятся в структуре соединения, то их значения позиций уменьшаются по модулю. Тем самым обеспечивается подвод к следующей возможной синхронной позиции (т.н. *относительная синхронизация*: например, следующая впадина между зубьями). Если для ведомой оси не дано "Разрешение наложения ведомой оси", сигнал интерфейсов DB(30 +номер оси), DBX 26 бит 4, то движение на синхронную позицию не выполняется. Вместо этого программа останавливается на кадре EGONSYN и самоудаляющаяся ошибка 16771 сигнализируется до тех пор, пока в.у. сигнал не будет установлен.

Вариант 3:

Расстояние между зубьями (градус) получается из: $360 \cdot Z_i/N_i$. В том случае, если ведомая на момент вызова остановлена, то поведение оптимизации по ходу идентично таковому оптимизации по времени.

Если ведомая ось уже перемещается, с помощью NTGP синхронизация со следующей впадиной между зубьями выполняется независимо от текущей скорости ведомой оси. Если ведомая ось уже перемещается, с помощью NTGT синхронизация со следующей впадиной между зубьями выполняется в зависимости от текущей скорости ведомой оси. Для этого при необходимости выполняется и торможение оси.

Таблицы кривых

Если для одной из ведущих осей используется **таблица кривых**, то:

Ni	Знаменатель коэффициента связи линейных соединений должен быть установлен на 0. (знаменатель 0 был бы недопустим для линейных соединений). Знаменатель ноль указывает СЧПУ на то, что
Zi	должен быть интерпретирован как номер используемой таблицы кривых. Таблица кривых с указанным номером на момент включения должна быть уже определена.
LAi	Указание ведущей оси соответствует указанию ведущей оси при соединении через коэффициент связи (линейное соединение).

Дополнительные указания по использованию таблиц кривых и каскадированию электронных редукторов и их синхронизации см.:

Литература:

Описание функций "Специальные функции"; Соединения осей и ESR (M3), глава "Буксировка и соединение по главному значению".

Поведение электронного редуктора при Power On, RESET, смене режимов работы, поиске кадра

- После Power On **нет** активного соединения.
- Активные соединения сохраняются после RESET и смене режимов работы.
- При поиске кадра команды для включения, удаления, определения электронного редуктора не выполняются и не собираются, а пропускаются.

Системные переменные электронного редуктора

С помощью системных переменных электронного редуктора программа обработки детали может определять текущее состояние структуры осей электронного редуктора и при необходимости реагировать на них.

Системные переменные электронного редуктора обозначены следующим образом:

\$AA_EG_ ...

или

\$VA_EG_ ...

Литература:

Справочник по системным переменным

13.4.3 Выключение электронного редуктора (EGOFS, EGOFC)

Для выключения активной структуры осей электронного редуктора существует 3 варианта.

Программирование

Вариант 1:

Синтаксис	Значение
EGOFS (ведомая ось)	Электронный редуктор отключается. Ведомая ось затормаживается до состояния покоя. Вызов запускает остановку предварительной обработки.

Вариант 2:

Синтаксис	Значение
EGOFS (ведомая ось, ведущая ось1, ..., ведущая ось5)	Такое параметрирование команды позволяет выборочно устранять влияние отдельных ведущих осей на движение ведомой оси.

Должна быть указана минимум одна ведущая ось. Воздействие указанных ведущих осей на ведомую ось целенаправленно отключается. Вызов запускает остановку предварительной обработки. Если еще остаются активные ведущие оси, то ведомая ось продолжает свое движение под их влиянием. Если все влияния ведущих осей отключаются таким способом, то ведомая ось затормаживается до состояния покоя.

Вариант 3:

Синтаксис	Значение
EGOFC (ведомый шпиндель)	Электронный редуктор отключается. Ведомый шпиндель продолжает движение с актуальным на момент времени отключения числом оборотов/скоростью. Вызов запускает остановку предварительной обработки.

Примечание

Этот вариант разрешен только для шпинделей.

13.4.4 Удаление определения электронного редуктора (EGDEL)

Необходимо отключить структуру осей электронного редуктора перед удалением его определения.

Программирование

Синтаксис	Значение
EGDEL (ведомая ось)	Определение соединения структуры осей удаляется. Снова появляется возможность повторного определения других структур осей с EGDEF до достижения макс. числа активированных одновременно структур осей. Вызов запускает остановку предварительной обработки.

13.4.5 Окружная подача (G95) / электронный редуктор (FPR)

С помощью команды FPR и ведомая ось электронного редуктора может быть указана в качестве определяющей подачу оси окружной подачи. На этот случай имеется следующее поведение:

- Подача зависит от заданной скорости ведомой оси электронного редуктора.
- Заданная скорость вычисляется из скоростей ходовых винтов и ведущих осей модуло (которые не являются траекторными осями) и их соответствующих коэффициентов связи.
- Компоненты скорости линейных ведущих осей или не модуло ведущих осей и наложенные движения ведомой оси не учитываются.

13.5 Синхронный шпиндель

В синхронном режиме существует один ходовой винт (LS) и один ведомый шпиндель (FS), т.н. **синхронная шпиндельная пара**. Ведомый шпиндель следует при активном соединении (синхронный режим) за движениями ходового винта согласно установленной функциональной связи.

Синхронные шпиндельные пары могут быть фиксировано сконфигурированы для любого станка посредством спец. для канала машинных данных, или определены спец. для приложения через программу обработки детали ЧПУ.. Для каждого канала ЧПУ возможно одновременное использование до 2 синхронных шпиндельных пар.

Соединение из программы обработки детали может быть

- определено или изменено
- включено
- отключено
- удалено

Кроме этого, в зависимости от версии ПО, можно

- ожидать условия синхронного хода
- изменить поведение смены кадра
- в качестве типа соединения выбрать либо соединение по заданному значению, либо соединение по фактическому значению или задать угловое смещение между ходовым винтом и ведомым шпинделем
- при включении соединения применить предшествующее программирование ведомого шпинделя
- исправить либо измеренную, либо уже известную погрешность синхронного хода

13.5.1 Синхронный шпиндель: Программирование (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

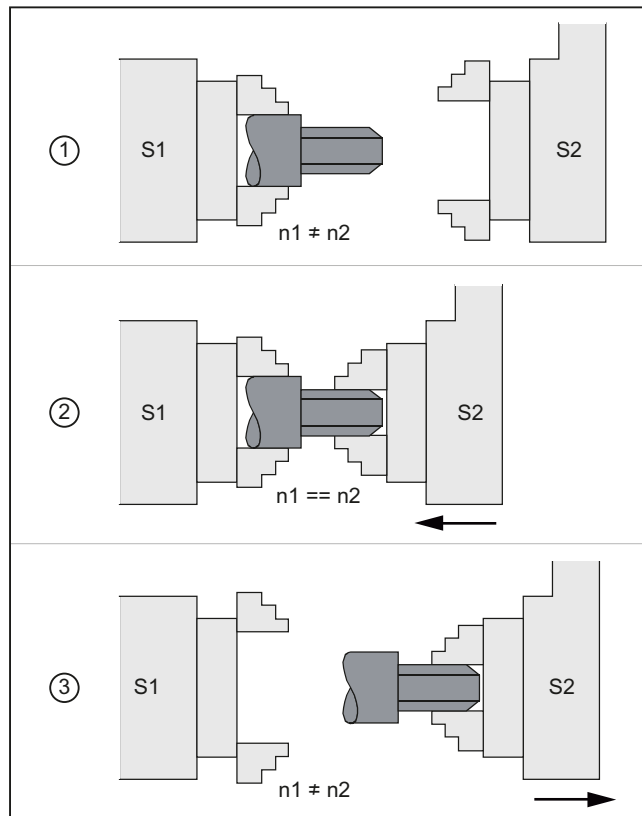
Функция "Синхронный шпиндель" позволяет выполнять синхронное по частоте вращения перемещение ведомого (FS) и ведущего шпинделя (LS) с программируемым передаточным отношением.

Функция предлагает следующие режимы:

- Синхронность по частоте вращения ($n_{FS} = n_{LS}$)
- Синхронность по положению ($\phi_{FS} = \phi_{LS}$)
- Синхронность по положению с угловым смещением ($\phi_{FS} = \phi_{LS} + \Delta\phi$)

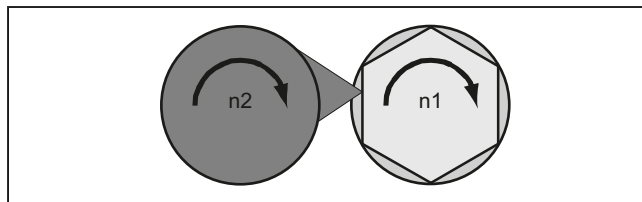
Примеры использования:

- Передача детали на лету, например, обработка задней стороны, передаточное отношение: 1:1



- ① Синхронизация частоты вращения
- ② Передача детали
- ③ Обработка "задней стороны"

- Многогранная обработка (полигональная токарная обработка), синхронная частота вращения, передаточное отношение: $n_1:n_2$



Синтаксис

```
COUPDEF (<FS>, <LS>, <ZFS>, <NLS>, <смена кадра>, <тип соединения>)
COUPON (<FS>, <LS>, <POSFS>)
COUPONC (<FS>, <LS>)
COUPOF (<FS>, <LS>, <POSFS>, <POSLS>)
COUPOFS (<FS>, <LS>)
COUPOFS (<FS>, <LS>, <POSFS>)
COUPRES (<FS>, <LS>)
```

COUPDEL (<FS>, <LS>)
WAITC (<FS>, <смена кадров>, <LS>, <смена кадров>)

Примечание

Сокращенная форма записи

Для операторов COUPOF, COUPOFS, COUPRES и COUPDEL возможна сокращенная форма записи без указания ведущего шпинделя.

Объяснение

COUPDEF:	Спец. для пользователя определение/изменение соединения
COUPON:	Включить соединение. Исходя из текущей частоты вращения, ведомый шпиндель синхронизируется с ведущим шпинделем
COUPONC:	При включении применить соединение с прежним программированием M3 S... или M4 S... Дифф. частота вращения ведомого шпинделя применяется сразу же.
COUPOF:	Выключить соединение. - с мгновенной сменой кадров: COUPOF (<S2>, <S1>) - Смена кадров только после перехода через позицию(и) отключения <POSFS> или <POSLS>: COUPOF (<S2>, <S1>, <POSFS>) COUPOF (<S2>, <S1>, <POSFS>, <POSLS>)
COUPOFS:	Отключение соединения с остановкой ведомого шпинделя. Макс. быстрая смена кадра с мгновенной сменой кадра: COUPOFS (<S2>, <S1>) Смена кадра только после перехода позиции отключения: COUPOFS (<S2>, <S1>, <POSFS>)
COUPRES:	Сбросить параметры соединения на сконфигурированные MD и SD
COUPDEL:	Удалить определенное пользователем соединение
WAITC:	Ожидать условия синхронного хода (NOC на IPO при смене кадра отменяются)
<FS>:	Обозначение ведомого шпинделя
Опциональные параметры:	
<LS>:	Обозначение ведущего шпинделя Указание с номером шпинделя: например, S2, S1
<ZFS>, <NLS>:	Передаточное отношение между FS и LS. <ZFS> / <NLS> = числитель / знаменатель Предустановка: <ZFS> / <NLS> = 1.0 ; указание знаменателя не обязательно

<смена кадров>:	Параметры смены кадра	
	Смена кадра выполняется:	
	"NOC"	сразу же
	"FINE"	при достижении "Синхронного хода точного"
	"COARSE"	при достижении "Синхронного хода грубого"
	"IPOSTOP"	при достижении IPOSTOP, т. е. после синхронного хода со стороны заданного значения (предустановка)
<тип соединения>:	Параметры смены кадра действуют модально	
	Тип соединения: Соединение между FS и LS	
	"DV"	Соединение по заданному значению (предустановка)
	"AV"	Соединение по фактическому значению
	"VV"	Соединение по скорости
	Тип соединения действует модально.	
<POSFS>:	Угловое смещение между ведущим и ведомым шпинделем	
	Диапазон значений:	0°... 359,999°
<POSFS>, <POSLS>:	Позиции отключения ведомого и ведущего шпинделя	
	"Смена кадра разрешается после перехода через POS _{FS} , POS _{LS} "	
<POSFS>, <POSLS>:	Диапазон значений:	
	0°... 359,999°	

Примеры

Работа с ведущим и ведомым шпинделем

Программный код	Комментарий
N05 M3 S3000 M2=4 S2=500	Ведущий шпиндель = мастер-шпиндель = шпиндель 1 Ведомый шпиндель = шпиндель 2 Ведущий шпиндель вращается с 3000 об/мин, ведомый шпиндель с 500 об/мин.
N10 COUPDEF(S2,S1,1,1,"NOC","DV")	Определение соединения (возможно и конфигурирование).
...	
N70 SPCON	Перевести ведущий шпиндель в управление по положению (соединение по заданному значению)
N75 SPCON(2)	Перевести ведомый шпиндель в управление по положению.
N80 COUPON(S2,S1,45)	Соединение на позицию смещения = 45 градусов "на лету".
...	
N200 FA[S2]=100	Скорость позиционирования = 100 градусов/мин
N205 SPOS[2]=IC(-90)	Движение с наложением 90 градусов в отрицательном направлении.
N210 WAITC(S2,"Fine")	Ожидание синхронного хода "точного".
N212 G1 X... Y... F...	Обработка
...	
N215 SPOS[2]=IC(180)	Движение с наложением 180 градусов в положительном направлении.
N220 G4 S50	Время ожидания = 50 оборотов мастер-шпинделя

Программный код	Комментарий
N225 FA[S2]=0	Активировать сконфигурированную скорость (MD) .
N230 SPOS[2]=IC(-7200)	20 оборотов. Двигаться со сконфигурированной скоростью в отрицательном направлении.
...	
N350 COUPOF(S2,S1)	Разъединение "на лету", S=S2=3000
N355 SPOSA[2]=0	Остановить FS на нуле градусов.
N360 G0 X0 Y0	
N365 WAITS(2)	Ожидать шпиндель 2.
N370 M5	Остановить FS.
N375 M30	

Программирование дифф. скорости

Программный код	Комментарий
	Ведущий шпиндель = мастер-шпиндель = шпиндель 1
	Ведомый шпиндель = шпиндель 2
N01 M3 S500	Ведущий шпиндель вращается с 500 об/мин.
N02 M2=3 S2=300	Ведомый шпиндель вращается с 300 об/мин.
...	
N10 G4 F1	Время ожидания мастер-шпинделя.
N15 COUPDEF(S2,S1,-1)	Коэффициент связи с передаточным отношением -1:1
N20 COUPON(S2,S1)	Активировать связь. Частота вращения ведомого шпинделя получается из частоты вращения ведущего шпинделя и коэффициента связи.
...	
N26 M2=3 S2=100	Программирование дифф. частоты вращения

Примеры приема движения по дифф. скорости

1. Включить соединение с COUPON при предшествующем программировании ведомого шпинделя

Программный код	Комментарий
	Ведущий шпиндель = мастер-шпиндель = шпиндель 1
	Ведомый шпиндель = шпиндель 2
N05 M3 S100 M2=3 S2=200	Ведущий шпиндель вращается с 100 об/мин, ведомый шпиндель с 200 об/мин.
N10 G4 F5	Время ожидания = 5 секунд мастер-шпинделя
N15 COUPDEF(S2,S1,1)	Передаточное отношение FS к LS равно 1,0 (предустановка).
N20 COUPON(S2,S1)	Соединение с ведущим шпинделем "на лету".
N10 G4 F5	Ведомый шпиндель вращается с 100 об/мин.

2. Включить соединение с COUPONC при предшествующем программировании ведомого шпинделя

Программный код	Комментарий
	Ведущий шпиндель = мастер-шпиндель = шпиндель 1

Программный код	Комментарий
N05 M3 S100 M2=3 S2=200	Ведомый шпиндель = шпиндель 2 Ведущий шпиндель вращается с 100 об/мин, ведомый шпиндель с 200 об/мин.
N10 G4 F5	Время ожидания = 5 секунд мастер-шпинделя
N15 COUPDEF(S2,S1,1)	Передаточное отношение FS к LS равно 1,0 (предустановка).
N20 COUPONC(S2,S1)	Соединение с ведущим шпинделем "на лету" и применить прежнюю частоту вращения к S2.
N10 G4 F5	S2 вращается с 100об/мин + 200об/мин = 300об/мин

3. Включить соединение с COUPON при стоящем ведомом шпинделе

Программный код	Комментарий
N05 SPOS=10 SPOS[2]=20	Ведущий шпиндель = мастер-шпиндель = шпиндель 1 Ведомый шпиндель = шпиндель 2 Ведомый шпиндель S2 в режиме позиционирования.
N15 COUPDEF(S2,S1,1)	Передаточное отношение FS к LS равно 1,0 (предустановка).
N20 COUPON(S2,S1)	Соединение с ведущим шпинделем "на лету".
N10 G4 F1	Соединение закрывается, S2 останавливается на 20 градусах.

4. Включить соединение с COUPONC при стоящем ведомом шпинделе

Примечание

Режим позиционирования или осевой режим

Если ведомый шпиндель перед соединением находится в режиме позиционирования или осевом режиме, то поведение ведомого шпинделя при COUPON (<FS>, <LS>) и COUPONC (<FS>, <LS>) идентично.

Примечание

Ведущий шпиндель в осевом режиме

Если ведущий шпиндель перед определением соединения находится в осевом режиме, то и после включения соединения действует предельное значение скорости из машинных данных:

MD32000 \$MA_MAX_AX_VELO (макс. скорость оси)

Во избежание такого поведения, ось перед определением соединения должна быть переведена в шпиндельный режим (M3 S... или M4 S...).

Дополнительная информация

Сконфигурированное соединение

В случае сконфигурированного соединения LS и FS определяются через машинные данные. Сконфигурированные шпиндели не могут быть изменены в программе

обработки детали. Параметрирование соединения возможно в программе обработки детали с COUPDEF (условие: защита записи не установлена).

Определенное пользователем соединение

С помощью COUPDEF можно заново определить или изменить соединение в программе обработки детали. Если соединение уже активно, то перед определением нового соединения оно должно быть сначала удалено с COUPDEL.

Соединение полностью определяется через:

COUPDEF (<FS>, <LS>, <ÜFS>, <ÜLS>, параметры смены кадра, тип соединения)

Ведомый шпиндель (FS) и ведущий шпиндель (LS)

С именами осей для FS и LS соединение определяется однозначно. Имена осей должны программироваться с каждым оператором COUPDEF. Другие параметры соединения действуют модально и должны повторно программироваться только после их изменения.

Пример:

```
COUPDEF (S2, S1)
```

Передаточное отношение

Передаточное отношение указывается как отношение по скорости между FS и LS:

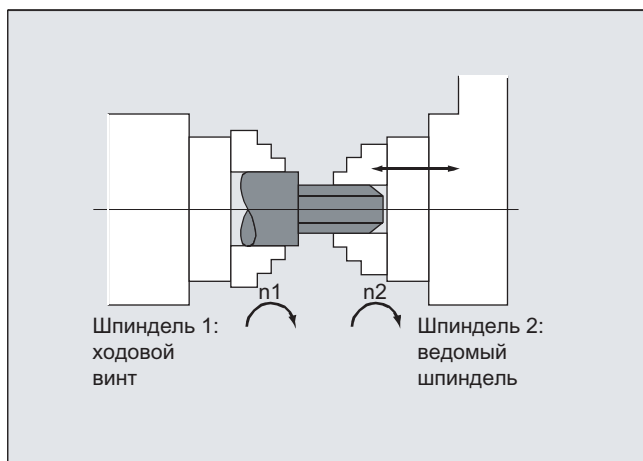
Ведомый шпиндель / ведущий шпиндель = числитель / знаменатель

Числитель должен быть запрограммирован. Программирование знаменателя не требуется. Тогда для знаменателя устанавливается значение по умолчанию 1.0.

Пример:

Ведомый шпиндель S2 и ведущий шпиндель S1, передаточное отношение = 1 / 1

```
COUPDEF (S2, S1, 1.0)
```



Примечание

Передаточное отношение может изменяться и при включенном соединении и вращающихся шпинделях.

Параметры смены кадра NOC, FINE, COARSE, IPOSTOP

При программировании параметров смены кадров возможна следующая сокращенная форма записи:

- "NO": сразу же (предустановка)
- "FI": при достижении "Синхронного хода точного"
- "CO": при достижении "Синхронного хода грубого"
- "IP": при достижении IPOSTOP, т. е. после синхронного хода со стороны заданного значения

Тип соединения

Примечание

Тип соединения может изменяться только при отключенном соединении!

Включение синхронного режима COUPON, <POSFS>

- Включение соединения с любым угловым смещением между FS и LS:
 - COUPON (S2, S1)
 - COUPON (S2)
- Включение соединения с угловым смещением <POSFS>
<POSFS> относится к позиции 0° ведущего шпинделя в положительном направлении вращения
Диапазон значений <POSFS>: 0°... 359,999°
 - COUPON (S2, S1, 30)

Примечание

Угловое смещение может изменяться и при активном соединении.

Позиционирование ведомого шпинделя

И при включенном соединении синхронных шпинделей возможно позиционирование FS независимо от LS в диапазоне $\pm 180^\circ$.

- Позиционирование шпинделя FS с SPOS
Пример: SPOS[2]=IC(-90)
Дополнительную информацию по SPOS можно найти в:
Литература:
Руководство по программированию "Основы"

Дифф. скорость

Дифф. скорость возникает в режиме управления по скорости и активном соединении синхронных шпинделей через наложение со знаком скорости FS на основе движения LS и скорости FS на основе программирования шпинделя:

- Соединение синхронных шпинделей с COUPONC
- $S<FS>=<скорость> [M<FS>=<направление вращения>]$

Примечание

Граничные условия

- При направлении вращения M3 / M4 необходимо повторное программирование и скорости S . . .
- Наложения скорости шпинделя ($M<направление вращения> S<FS>$) через движение LS в соединении синхронных шпинделей COUPONC активируется только при разрешенном наложении.
- Динамика ведущего шпинделя должна быть ограничена настолько, чтобы при наложении ведомого шпинделя не были бы превышены его предельные значения динамики.

Дополнительную информацию по дифф. скорости см.:

Литература:

Описание функций "Дополнительные функции"; Синхронный шпиндель (S3)

Скорость, ускорение: FA, ACC, OVRA, VELOLIMA

Осевая скорость и ускорение ведомого шпинделя могут программироваться с:

- $FA[SPI(S<n>)]$ или $FA[S<n>]$ (осевая скорость)
- $ACC[SPI(S<n>)]$ или $ACC[S<n>]$ (осевое ускорение)
- $OVRA[SPI(S<n>)]$ или $OVRA[S<n>]$ (осевая процентовка)
- $VELOLIMA[SPI(S<n>)]$ или $VELOLIMA[S<n>]$ (осевое превышение или снижение скорости)

$S<n> = 1, 2, 3, \dots$ (номер ведомого шпинделя)

Литература:

Руководство по программированию "Основы"

Примечание

Понижение превышения макс. осевого рывка для шпинделей не действует.

Дополнительную информацию по осевой динамике см.:

Литература:

Описание функций - Дополнительные функции; Круговые оси (R2)

Программируемые параметры смена кадра WAITC

С помощью WAITC параметры смены кадра, к примеру, после изменения параметров соединения или процессов позиционирования, могут задаваться с различными условиями синхронного хода (грубый, точный, IPOSTOP). Если условия синхронного

хода не указаны, то действуют указанные при определении COUPDEF параметры смены кадра.

Примеры

- Ожидание достижения условия синхронного хода FINE для ведомого шпинделя S2 и COARSE для ведомого шпинделя S4: WAITC (S2, "FINE", S4, "COARSE")
- Ожидание достижения условия синхронного хода согласно COUPDEF: WAITC ()

Отключить соединение COUPOF

С помощью COUPOF могут быть заданы параметры отключения соединения:

- Отключение соединения с мгновенной сменой кадра:
 - COUPOF (S2, S1) (с указанием ведущего шпинделя)
 - COUPOF (S2) (с указанием ведущего шпинделя)
- Отключение соединения после перехода позиций отключения. Смена кадров происходит после перехода позиций отключения.
 - COUPOF (S2, S1, 150) (позиция отключения FS: 150°)
 - COUPOF (S2, S1, 150, 30) (позиция отключения FS: 150°, LS: 30°)

Отключение соединения с остановкой ведомого шпинделя COUPOFS

С помощью COUPOFS могут быть заданы параметры отключения соединения с остановкой ведомого шпинделя:

- Отключение соединения с остановкой ведомого шпинделя и мгновенной сменой кадра:
 - COUPOFS (S2, S1) (с указанием ведущего шпинделя)
 - COUPOFS (S2) (с указанием ведущего шпинделя)
- Отключение соединения после перехода позиций отключения с остановкой ведомого шпинделя. Смена кадров происходит после перехода позиций отключения.
 - COUPOFS (S2, S1, 150) (позиция отключения FS: 150°)

Удаление соединений COUPDEL

С COUPDEL соединение удаляется:

- COUPDEL (S2, S1) (с указанием ведущего шпинделя)
- COUPDEL (S2) (с указанием ведущего шпинделя)

Сброс параметров соединения COUPRES

С COUPRES активируются спараметрированные в машинных и установочных данных значения соединения:

- COUPRES (S2, S1) (с указанием ведущего шпинделя)
- COUPRES (S2) (с указанием ведущего шпинделя)

Системные переменные

- Актуальное состояние соединения ведомого шпинделя
Текущее состояние соединения ведомого шпинделя может быть считано в битовой кодировке через:

<значение> = \$AA_COUP_ACT [<FS>]

Бит	<значение>	Значение
-	0	нет активного соединения
2	4	активно соединение синхронных шпинделей
Указание - Все другие значения относятся к осевому режиму - Если шпиндель является ведомым шпинделем из нескольких соединений, в качестве значения состояния соединения всех соединений возвращается суммарное соединение.		

- Актуальное угловое смещение
Текущее угловое смещение ведомого шпинделя к ведущему шпинделю может быть считано через:

– \$AA_COUP_OFFS [<FS>] (угловое смещение со стороны заданного значения)

– \$VA_COUP_OFFS [<FS>] (угловое смещение со стороны фактического значения)

Пример использования

Коррекция разницы угловых смещений в программе ЧПУ после отмены режима слежения:

Разность угловых смещений = запрограммированное угловое смещение - системная переменная

Литература

Подробную информацию по системным переменным см.:

Справочник по параметрированию "Системные переменные"

13.6 Базовое соединение осей (CP...)

"Базовое соединение осей" это общая функция соединения, объединяющая в себе все свойства существующих типов соединений (совместное движение, соединение по главному значению, электронный редуктор и синхронный шпиндель).

Функция позволяет выполнять гибкое программирование:

- Пользователь может свободно выбирать необходимые для его приложения свойства соединения (модульный принцип).
- Каждое свойство соединения может быть запрограммировано по отдельности.
- Свойства определенного соединения (например, коэффициент связи) могут изменяться.
- Последующее использование других свойств соединения возможно.
- Базовая система ведомой оси (базовая кинематическая система или система координат станка) может программироваться.
- Определенные свойства соединения могут программироваться и в синхронных действиях.

Литература: Описание функций "Синхронные действия"

Примечание

Прежние вызовы соединений для совместного движения (TRAIL*), соединения по главному значению (LEAD*), электронного редуктора (EG*) и синхронного шпинделя (COUP*) продолжают поддерживаться через адаптивные циклы.

Обзор всех кодовых слов и свойств соединений

Следующая таблица предлагает обзор всех кодовых слов базового соединения осей и программируемых с их помощью свойств соединения:

Кодовое слово	Свойство соединения / значение	Синтаксис
CPDEF	Создание модуля сопряжения	CPDEF= (<FAx>)
CPDEL	Удаление модуля сопряжения	CPDEL= (<FAx>)
CPLA	Определение ведущей оси	CPLA [<FAx>] = (<LAx>)
CPLDEF	Определение ведущей оси и создание модуля сопряжения (также возможно с CPDEF + CPLA)	CPLDEF [<FAx>] = (<LAx>) или CPDEF= (<FAx>) CPLA [<FAx>] = (<LAx>)
CPLDEL	Удаление ведущей оси модуля сопряжения (также возможно с CPDEL + CPLA)	CPLDEL [<FAx>] = (<LAx>) или CPDEL= (<FAx>) CPLA [<FAx>] = (<LAx>)
CPON	Включение модуля сопряжения	CPON= (<FAx>)

13.6 Базовое соединение осей (CP...)

Кодовое слово	Свойство соединения / значение	Синтаксис
CPOF	Выключение модуля сопряжения	CPOF = (<FAx>)
CPLON	Включение ведущей оси модуля сопряжения	CPLON [<FAx>] = <LAx>
CPLOF	Выключение ведущей оси модуля сопряжения	CPLOF [<FAx>] = <LAx>
CPLNUM	Числитель коэффициента связи	CPLNUM [FAx, LAx] = <значение>
CPLDEN	Знаменатель коэффициента связи	CPLDEN [FAx, LAx] = <значение>
CPLCTID	Номер таблицы кривых	CPLCTID [FAx, LAx] = <значение>
CPLSETVAL	Источник соединения	CPLSETVAL [FAx, LAx] = "<источник соединения>"
		"<Источник соединения>":
		"CMDPOS" Соединение по заданному значению
		"CMDVEL" Соединение по скорости
CPFRS	Исходная система координат	CPFRS [FAx] = "<источник координат>"
		"<Источник координат>":
		"BCS" Базовая кинематическая система
		"MCS" Система координат станка
CPBC	Критерий смены кадров	CPBC [FAx] = "<критерий смены кадра>"
		"<Критерий смены кадра>":
		"NOC" Смена кадра выполняется независимо от состояния соединения.
		"IPOSTOP" Смена кадра при синхронном ходе со стороны заданного значения.
		"COARSE" Смена кадра при "Синхронном ходе грубом" со стороны фактического значения.
		"FINE" Смена кадра при "Синхронном ходе точном" со стороны фактического значения.
CPFPOS + CPON	Синхронная позиция ведомой оси при включении	CPON=FAx CPFPOS [FAx] = <значение>
CPLPOS + CPON	Синхронная позиция ведущей оси при включении	CPLPOS [FAx, LAx] = <значение>

Кодовое слово	Свойство соединения / значение	Синтаксис
CPFMSON	Режим синхронизации	CPFMSON [FAx] = "<режим синхронизации>"
		"<Режим синхронизации>":
		"CFAST" Соединение замыкается оптимально по времени.
		"CCOARSE" Соединение включается только в том случае, если требуемая по правилу связи позиция ведомой оси находится в области текущей позиции ведомой оси.
		"NTGT" Оптимизированный по времени подвод к следующей впадине между зубьями.
		"NTGP" Оптимизированный по пути подвод к следующей впадине между зубьями.
		"NRGT" Оптимизированный по времени подвод к следующему сегменту согласно отношению из числа ступеней к числу зубьев.
		"NRGP" Оптимизированный по пути подвод к следующему сегменту согласно отношению из числа ступеней к числу зубьев.
		"ACN" Только для круговых осей! Круговая ось подводится к синхронной позиции в отрицательном осевом направлении. Синхронизация происходит сразу же.
		"ACP" Только для круговых осей! Круговая ось подводится к синхронной позиции в положительном осевом направлении. Синхронизация происходит сразу же.
		"DCT" Только для круговых осей! Круговая ось подводится к запрограммированной синхронной позиции с оптимизацией по времени.

Кодовое слово	Свойство соединения / значение	Синтаксис		
				Синхронизация происходит сразу же.
			"DCP"	Только для круговых осей! Круговая ось подводится к запрограммированной синхронной позиции с оптимизацией по пути. Синхронизация происходит сразу же.
CPFMON	Поведение ведомой оси при включении	CPFMON [FAX] = "<поведение при включении>" "<Поведение при включении>":	"STOP"	Только для шпинделей! Активное движение ведомого шпинделя останавливается перед включением.
			"CONT"	Только для шпинделей и осей главного хода! Текущее движение ведомой оси/шпинделя берется в соединение как движение запуска.
			"ADD"	Только для шпинделей! Компоненты движения соединения действуют дополнительно к текущему наложенному движению, т. е. текущее движение ведомой оси/шпинделя сохраняется как наложенное движение.
CPFMOF	Поведение ведомой оси при полном выключении	CPFMOF [FAX] = "<поведение при выключении>" "<Поведение при выключении>":	"STOP"	Останов ведомой оси/шпинделя. Активное наложенное движение затормаживается до состояния покоя. После этого соединение размыкается.
			"CONT"	Только для шпинделей и осей главного хода! Ведомый шпиндель продолжает движение с актуальной на момент времени отключения частотой вращения/скоростью.
CPFPOS + CPOF	Позиция отключения ведомой оси при выключении	CPOF= (FAX) CPFPOS [FAX] =<значение>		

Кодовое слово	Свойство соединения / значение	Синтаксис		
CPMRESET	Поведение соединения при RESET	CPMRESET [FAx] = "<реакция на Reset>"		
		<Реакция на Reset>":	"NONE"	Текущее состояние соединения сохраняется.
			"ON"	Если соответствующий модуль сопряжения создан, то соединение включается. Активируются все определенные отношения ведущей оси. Это происходит и тогда, когда все или часть этих отношений ведущей оси уже активны, т. е. при полностью активированном соединении выполняется повторная синхронизация.
			"OF"	Активное наложенное движение затормаживается до состояния покоя. После этого соединение выключается. Если соответствующий модуль сопряжения был создан без явного определения (CPDEF), то модуль сопряжения удаляется. В ином случае он сохраняется, т. е. может использоваться и дальше.
			"OFC"	Возможно только для шпинделей! Ведомый шпиндель продолжает движение с актуальной на момент времени отключения частотой вращения/скоростью. Соединение выключается. Если соответствующий модуль сопряжения был создан без явного определения (CPDEF), то модуль сопряжения удаляется. В ином случае он сохраняется, т. е. может использоваться и дальше.
			"DEL"	Активное наложенное движение затормаживается до состояния покоя. После этого соединение деактивируется и удаляется.

Кодовое слово	Свойство соединения / значение	Синтаксис		
			"DELC"	Возможно только для шпинделей! Ведомый шпиндель продолжает движение с актуальной на момент времени отключения частотой вращения/скоростью. Соединение деактивируется и после удаляется.
CPMSTART	Поведение соединения при запуске программы обработки детали	CPMSTART [FАх] ="<поведение при запуске>"		
		"<Поведение при запуске>":	"NONE"	Текущее состояние соединения сохраняется.
			"ON"	Соединение включается. Активируются все определенные отношения ведущей оси. Это происходит и тогда, когда все или часть этих отношений ведущей оси уже активны, т. е. при полностью активированном соединении выполняется повторная синхронизация.
			"OF"	Соединение выключается. Если соответствующий модуль сопряжения был создан без явного определения (CPDEF), то модуль сопряжения удаляется. В ином случае он сохраняется, т. е. может использоваться и дальше.
		"DEL"	Соединение деактивируется и после удаляется.	
CPMPRT	Поведение соединения при запуске программы обработки детали под поиск кадра через тестирование программы	CPMPRT [FАх] ="<поведение при запуске>"		
		"<Поведение при запуске>":	см. CPMSTART	
CPLINTR	Значение смещения для входного значения ведущей оси	CPLINTR [FАх, LАх] =<значение>		
CPLINSC	Коэффициент масштабирования для входного значения ведущей оси	CPLINSC [FАх, LАх] =<значение>		
CPLOUTTR	Значение смещения для выходного значения соединения	CPLOUTTR [FАх, LАх] =<значение>		
CPLOUTSC	Коэффициент масштабирования для выходного значения соединения	CPLOUTSC [FАх, LАх] =<значение>		

13.6 Базовое соединение осей (CP...)

Кодовое слово	Свойство соединения / значение	Синтаксис		
CPSYNCOF	Пороговое значение для позиционного синхронного хода "грубого"	CPSYNCOF [FAx] = <значение>		
CPSYNFIP	Пороговое значение для позиционного синхронного хода "точного"	CPSYNFIP [FAx] =<значение>		
CPSYNCOF2	Второе пороговое значение для позиционного синхронного хода "грубого"	CPSYNCOF2 [FAx] =<значение>		
CPSYNFIP2	Второе пороговое значение для позиционного синхронного хода "точного"	CPSYNFIP2 [FAx] =<значение>		
CPSYNCOV	Пороговое значение для скоростного синхронного хода "грубого"	CPSYNCOV [FAx] =<значение>		
CPSYNFIV	Пороговое значение для скоростного синхронного хода "точного"	CPSYNFIV [FAx] =<значение>		
CPMBRAKE	Поведение ведомой оси при определенных сигналах и командах останова	CPMBRAKE [FAx] =<биткодированное значение>		
CPMVDI	Реакция ведомой оси на определенные интерфейсные сигналы ЧПУ/PLC	CPMVDI [FAx] =<биткодированное значение>		
CPMALARM	Блокировка вывода спец. относящихся к соединению аварийных сообщений	CPMALARM [FAx] =<биткодированное значение>		
CPSETTYPE	Тип соединения	CPSETTYPE [FAx] ="<тип соединения>"		
		"<тип соединения>":	"CP"	Свободная программируемость
			"TRAIL"	Тип соединения "совместное движение"
			"LEAD"	Тип соединения "соединение по главному значению"
			"EG"	Тип соединения "электронный редуктор"
	"COUP"	Тип соединения "синхронный шпиндель"		

FAx: ведомая ось/ведомый шпиндель

LAx: ведущая ось/ведущий шпиндель

Примечание

Свойства соединения, которые не программируются явно (в программе обработки детали или в синхронных действиях), активируются со своими стандартным настройкам.

В зависимости от настройки кодового слова CPSETTYPE вместо стандартных настроек (CPSETTYPE="CP") могут быть активированы и предустановленные свойства соединения.

Литература

Подробную информацию по базовым соединениям осей можно найти в:

- Описание функций "Специальные функции"; МЗ: Соединения осей, глава: "Базовое соединение осей"

13.7 Соединение Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

"Соединение Master/Slave" обеспечивает:

- Соединение Slave-осей с Master-осью в состоянии покоя участвующих осей.
- Соединение и разъединение **вращающихся**, управляемых по скорости шпинделей.
- Динамическое конфигурирование.

Примечание

Режим позиционирования

У осей и шпинделей в режиме позиционирования соединение замыкается и разъединяется только в состоянии покоя.

Синтаксис

```
MASLON(<Slave_1>,<Slave_2>,...)
MASLOF(<Slave_1>,<Slave_2>,...)
MASLOFS(<Slave_1>,<Slave_2>,...)
```

Динамическое конфигурирование:

```
MASLDEF(<Slave_1>,<Slave_2>,...,<Master>)
MASLDEL(<Slave_1>,<Slave_2>,...)
```

Объяснение

MASLON:	Включение временного соединения Master/Slave	
	<Slave_x>,...:	Slave-ось 1 ... n
MASLOF:	Выключение активного соединения Master/Slave	
	<Slave_1>,...:	Slave-ось 1 ... n
MASLOFS:	Разрыв соединения Master/Slave и автоматическое затормаживание Slave-шпинделей (см. указание "Поведение шпинделей при соединении в режиме управления по скорости"!)	
	<Slave_1>,...:	Slave-ось 1 ... n
MASLDEF:	Создание/изменение структуры Master/Slave из программы обработки детали	
	<Slave_1>,...:	Slave-ось 1 ... n
	<Master>:	Master-ось
MASLDEL:	Разорвать соединение Master/Slave и удалить определение структуры	
	<Slave_1>,...:	Slave-ось 1 ... n
	Примечание: Сконфигурированные в машинных данных определения Master/Slave сохраняются.	

Примечание**Поведение шпинделей при соединении в режиме управления по скорости**

Для шпинделей в режиме управления по скорости поведение при соединении MASLON, MASLOF, MASLOFS и MASLDEL определяется явно через следующие машинные данные:

MD37263 \$MA_MS_SPIND_COUPLING_MODE

В стандартной настройке с MD 37263 = 0 соединение и разъединение осей Slave осуществляется исключительно в состоянии покоя участвующих осей. MASLOFS соответствует MASLOF.

При MD37263 = 1 оператор соединения выполняется сразу же, т. е. и при движении. Соединение при MASLON сразу же замыкается, а при MASLOFS или MASLOF сразу же размыкается. Вращающиеся при этом Slave-шпиндели сохраняют при MASLOF свои скорости до повторного программирования скорости. При MASLOFS, напротив, они автоматически затормаживаются.

Примечание

При MASLOF/MASLOFS не явная остановка предварительной обработки отсутствует. Из-за отсутствия остановки предварительной обработки системные переменные \$P до момента повторного программирования не выводят для осей Slave обновленных значений.

Примечание

Для Slave-осей фактическое значение через PRESETON может быть синхронизировано с тем же значением Master-оси. Для этого необходимо на короткое время отключить постоянное соединение Master/Slave, чтобы установить фактическое значение не реферированной Slave-оси при POWER ON на значение Master-оси. После этого снова восстанавливается постоянное соединение.

Постоянное соединение Master/Slave активируется следующей настройкой MD:

MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE = 1

Она не влияет на языковые команды временного соединения.

Примеры**Пример 1: Установка фактического значения для Slave-оси соединения Master/Slave**

При постоянном соединении Master/Slave с помощью PRESETON фактическое значение Slave-оси устанавливается на значение Master-оси.

Программный код	Комментарий
\$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; Выключение постоянного соединения Slave-оси
NEWCONF	; Активация изменения машинных данных
STOPRE	
MASLOF (Y1)	; Выключение временного соединения

13.7 Соединение Master/Slave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

Программный код	Комментарий
PRESETON (AX2, \$VA_IM(M_AX))	; Фактическое значение Slave-оси = фактическое значение Master-оси
\$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	; Включение постоянного соединения Slave-оси
NEWCONF	; Активация изменения машинных данных

Пример 2: Динамическое конфигурирование соединения Master/Slave

Для того, чтобы соединение после вращения осевого контейнера, могло бы быть замкнуто с другим шпинделем, прежде необходимо разъединить старое соединение, удалить конфигурацию и сконфигурировать новое соединение.

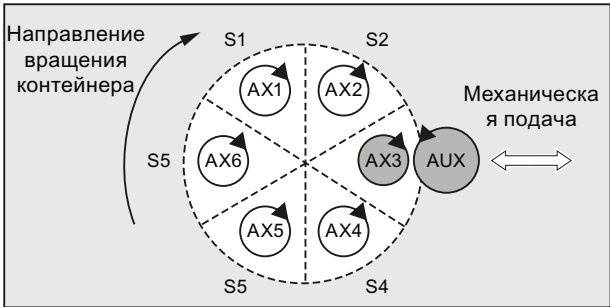


Рисунок 13-1 Перед поворотом осевого контейнера

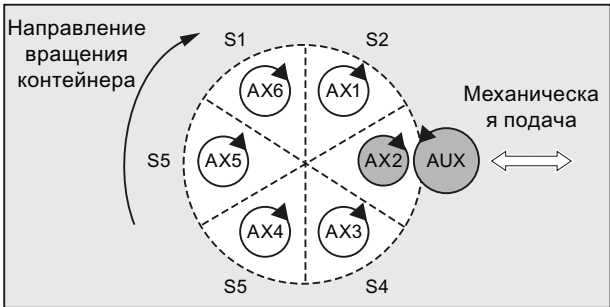


Рисунок 13-2 После поворота осевого контейнера на один слот

Программный код	Комментарий
MASLDEF (AUX, S3)	; AUX: Slave, S3: Master = AX3
MASLON (AUX)	; Соединение вкл
M3=3 S3=4000	; Поворот Master
MASLDEL (AUX)	; Разрыв и удаление соединения
AXCTSWE (CT1)	; Разрешение поворота осевого контейнера
MASLDEF (AUX, S3)	; AUX: Slave, S3: Master = AX2

Литература

- Описание функций "Специальные функции", глава "ТЕ3: Соединение по скорости/моменту вращения, Master-Slave"
- Описание функций "Дополнительные функции", глава "В3: Децентрализованные системы - только 840D sl" > "NCU-Link" > "Осевой контейнер"

Синхронные действия

14.1 Определение синхронного действия

Синхронное действие определяется в кадре программы обработки детали. В этом кадре не должно быть запрограммировано других команд, не являющихся составной частью синхронного действия.

Синхронное действие состоит из следующих компонентов:

Действительность, идент. номер. (опционально)	Условная часть (опционально)			Операционная часть		
	Частота	G-команда (опционально)	Условие	Кодовое слово	G-команда (опционально)	Операции
--- ¹⁾ ID=<Nr> IDS=<Nr>	--- ¹⁾ WHENEVER FROM WHEN EVERY	G...	Логическое выражение	DO	G...	Операция 1 ... Операция n

¹⁾ не запрограммировано

Синтаксис

```
DO <операция 1> ... <операция n>
<частота> [<G-функция>] <условие> DO <операция 1> ... <операция n>
ID=<Nr> <частота> [<G-функция>] <условие> DO <операция 1> ... <операция n>
IDS=<Nr> <частота> [<G-функция>] <условие> DO <операция 1> ... <операция n>
```

Литература

Подробное описание функциональности синхронных действий можно найти в:
Описание функций "Синхронные действия"

Качание

15.1 Асинхронное качание (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Качающаяся ось выполняет поступательно-возвратные движения между двумя точками возврата 1 и 2 с заданной подачей до отключения маятникового движения.

При маятниковом движении возможна любая интерполяция других осей. Через движение по траектории или с помощью позиционирующей оси можно достичь непрерывной подачи. Но при этом **связь** между маятниковым движением и движением подачи отсутствует.

Свойства асинхронного качания

- Асинхронное качание действует спец. для оси за границы кадра.
- Через программу обработки детали обеспечивается синхронное с кадром включение маятникового движения.
- Общая интерполяция нескольких осей и наложение участков качания невозможны.

Программирование

Через следующие команды возможно соответствующее обработке программы ЧПУ включение и управление асинхронным качанием из программы обработки детали.

Запрограммированные значения синхронно с кадром на главном ходе вносятся в соответствующие установочные данные и действуют до следующего изменения.

Синтаксис

```
OSP1 [<ось>]=<значение>  OSP2 [<ось>]=<значение>
OST1 [<ось>]=<значение>  OST2 [<ось>]=<значение>
FA [<ось>]=<значение>
OSCTRL [<ось>]=(<опция установки>,<опция сброса>)
OSNSC [<ось>]=<значение>
OSE [<ось>]=<значение>
OSB [<ось>]=<значение>
OS [<ось>]=1
OS [<ось>]=0
```

Объяснение

<ось>:	Имя качающейся оси		
OS:	Включение / выключение качания		
	Значение:	1	Включить качание
		0	Выключить качание
OSP1:	Определить позицию точки возврата 1		

OSP2:	Определить позицию точки возврата 2 Примечание: Если активно инкрементальное перемещение, то позиция вычисляется инкрементально к последней запрограммированной в программе ЧПУ соответствующей позиции возврата.		
OST1:	Определить время задержки в точке возврата 1 в [сек]		
OST2:	Определить время задержки в точке возврата 2 в [сек]		
	<значение >:	-2	Интерполяция продолжается без ожидания точного останова
		-1	Ожидание точного останова грубого
		0	Ожидание точного останова точного
		>0	Ожидание точного останова точного с последующим соблюдением указанного времени задержки Примечание: Единица для времени задержки идентична запрограммированному через G4 времени задержки.
FA:	Определить скорость подачи В качестве скорости подачи действует определенная скорость подачи позиционирующей оси. Если скорость подачи не определена, то действует зафиксированное в машинных данных значение.		

15.1 Асинхронное качание (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

OSCTRL:	Указать опции установки и сброса	
	Опциональные значения 0 - 3 кодируют поведение в точках возврата при отключении. Можно выбрать один из вариантов 0 - 3. Прочие установки при необходимости могут комбинироваться с выбранным вариантом. Несколько опций соединяются символом плюс (+).	
	<значение >:	0 При отключении маятникового движения остановиться в следующей точке возврата (предустановка) Примечание: Возможно только через сброс значений 1 и 2.
		1 При отключении маятникового движения остановка в точке возврата 1
		2 При отключении маятникового движения остановка в точке возврата 2
		3 При отключении маятникового движения не выполнять подвод к точке возврата, если ходы выхаживания не запрограммированы
		4 После выхаживания подвод к конечной позиции
		8 Если маятниковое движение отменяется через стирание остатка пути, то после должны быть выполнены ходы выхаживания и при необходимости переход к конечной позиции.
		16 Если маятниковое движение отменяется через стирание остатка пути, то, как при отключении, должен быть выполнен переход к соответствующей позиции возврата.
		32 Измененная подача активна только со следующей точки возврата
		64 FA равна 0, FA = 0: наложение хода активно FA отлична от 0, FA <> 0: наложение скорости активно
		128 Для круговой оси DC (кратчайший путь)
		256 Ход выхаживания выполняется как двойной ход (стандарт) 1=Ход выхаживания выполняется как одиночный ход.
		512 Сначала подвод в стартовую позицию
OSNSC:	Определить число ходов выхаживания	
OSE:	Определить конечную позицию (в WCS), к которой должен быть выполнен подвод после отключения качания Примечание: При программировании "OSE" для "OSCTRL" не явно начинает действовать опция 4.	
OSB:	Определить стартовую позицию (в WCS), в которую должен быть выполнен подвод перед включением качания Подвод в стартовую позицию выполняется до точки возврата 1. Если стартовая позиция совпадает с позицией возврата 1, то следующим шагом выполняется подвод в позицию возврата 2. При достижении стартовой позиции время задержки не действует, даже если стартовая позиция совпадает с позицией возврата 1, вместо этого выполняется ожидание точного останова точного. Установленное условие точного останова соблюдается. Примечание: Для подвода в стартовую позицию в установочных данных SD43770 \$SA_OSCILL_CTRL_MASK должен быть установлен бит 9.	

Примеры

Пример 1: Качающаяся ось должна качаться между двумя точками возврата

Качающаяся ось Z должна качаться между позицией 10 и 100. Подвод к точке возврата 1 с точным остановом точным, к точке возврата 2 с точным остановом грубым. Подача для качающейся оси должна составлять 250. В конце обработки должны быть выполнены 3 хода выхаживания и осуществлен подход к конечной позиции 200 с качающейся осью. Подача для оси подачи должна составлять 1, конец подачи в направлении X должен быть достигнут на позиции 15.

Программный код	Комментарий
WAITP (X, Y, Z)	; исходное положение.
G0 X100 Y100 Z100	; переключение в режим позиционирующей оси.
WAITP (X, Z)	
OSP1[Z]=10 OSP2[Z]=100	; точка возврата 1, точка возврата 2.
OSE[Z]=200	; конечная позиция.
OST1[Z]=0 OST2[Z]=-1	; время ожидания в U1: точный останов точный ; время ожидания в U2: точный останов грубый
FA[Z]=250 FA[X]=1	; подача качающейся оси, подача оси подачи.
OSCTRL[Z]=(4,0)	; установочные опции.
OSNSC[Z]=3	; 3 хода выхаживания.
OS[Z]=1	; запустить качание.
WHEN \$A_IN[3]==TRUE DO DELDTG(X)	; стирание остатка пути.
POS[X]=15	; исходное положение оси X
POS[X]=50	; конечное положение оси X.
OS[Z]=0	; остановить качание.
M30	

Примечание

Последовательность команд от "OSP1[Z]=..." до "OSNCS[Z]=..." может быть запрограммирована и в одном кадре.

Пример 2: Качание с изменением позиции возврата Online

Необходимые для асинхронного качания установочные данные могут быть установлены в программе обработки детали.

Если запись в установочные данные осуществляется в программе обработки детали напрямую, то изменение активируется уже на момент предварительной обработки. Синхронное поведение может быть достигнуто через остановку предварительной обработки (STOPRE).

Программный код	Комментарий
\$SA_OSCILL_REVERSE_POS1[Z] = -10	
\$SA_OSCILL_REVERSE_POS2[Z] = 10	
G0 X0 Z0	
WAITP (Z)	

15.1 Асинхронное качание (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Программный код	Комментарий
ID=1 WHENEVER \$AA_IM[Z] < \$\$AA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[X]=0	; Если фактическое значение качающейся оси превысило точку возврата, то ось подачи останавливается.
ID=2 WHENEVER \$AA_IM[Z] < \$\$AA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; включить качание.
OS[Z]=0	; отключить качание.
M30	

Дополнительная информация

Качающаяся ось

Для качающейся оси действует:

- Любая ось может использоваться как качающаяся ось.
- Одновременно может быть активно несколько качающихся осей (макс.: число позиционирующих осей).
- Для качающейся оси всегда активна линейная интерполяция G1, независимо от актуальной действующей в программе команды G.

Качающаяся ось может:

- Быть входной осью для динамической трансформации
- Быть ведущей осью для осей Gantry и буксируемых осей
- перемещаться:
 - без ограничения рывка "BRISK" или
 - с ограничением рывка "SOFT" или
 - с ломаной характеристикой ускорения (как позиционирующие оси)

Точки возврата качания

При определении позиций качания учитывать актуальные смещения:

- Абсолютное указание
"OSP1[Z]=<значение>"
Позиция точки возврата = сумма смещений + запрограммированное значение
- Относительное указание
"OSP1[Z]=IC(<значение>)"
Позиция точки возврата = точка возврата 1 + запрограммированное значение

Пример:

Программный код
N10 OSP1[Z] = 100 OSP2[Z] = 110
...
N40 OSP1[Z] = IC(3)

WAITP

Если необходимо качание с гео-осью, то она должна быть разрешена с "WAITP" для качания.

После завершеного качания с помощью "WAITP" качающаяся ось снова вносится как позиционирующая ось и может снова использоваться обычно.

Качание с синхронными действиями движения и временем задержки

По истечении установленного времени задержки при качании выполняется внутренняя смена кадра (видна по новым остаточным путям осей). При смене кадра проверяется функция отключения. При этом функция отключения определяется согласно выполненной установке СЧПУ для процесса движения (OSCTRL). *Этой характеристикой в функции времени можно управлять через процентовку подачи.*

При определенных обстоятельствах, после выполняется еще один ход качания, прежде чем будут запущены ходы выхаживания или выполнен подвод к конечной позиции. *При этом создается впечатление, что изменяется характеристика отключения. Но это не так.*

15.2 Управляемое через синхронные действия качание (OSCILL)

При этом типе качания движение подачи разрешено только на точках возврата или внутри определенных областей возврата.

В зависимости от требований, при подаче маятниковое движение может быть

- продолжено или
- остановлено до полного выполнения подачи.

Синтаксис

1. Определение параметров для качания
2. Определение синхронных действий движения
3. Согласование осей, определение подачи

Объяснение

OSP1 [<качающаяся ось>]=	Позиция точки возврата 1
OSP2 [<качающаяся ось>]=	Позиция точки возврата 2
OST1 [<качающаяся ось>]=	Время задержки в точке возврата 1 в секундах
OST2 [<качающаяся ось>]=	Время задержки в точке возврата 2 в секундах
FA [<качающаяся ось>]=	Подача качающейся оси
OSCTRL [<качающаяся ось>]=	Опции установки или сброса
OSNSC [<качающаяся ось>]=	Число ходов выхаживания
OSE [<качающаяся ось>]=	Конечная позиция
WAITP (<качающаяся ось>)	Разрешить ось для качания

Согласование осей, подача

OSCILL [<качающаяся ось>]=(<ось подачи 1>,<ось подачи 2>,<ось подачи 3>)

POSP [<ось подачи>]=(<конечная позиция>,<длина фрагмента>,<режим>)

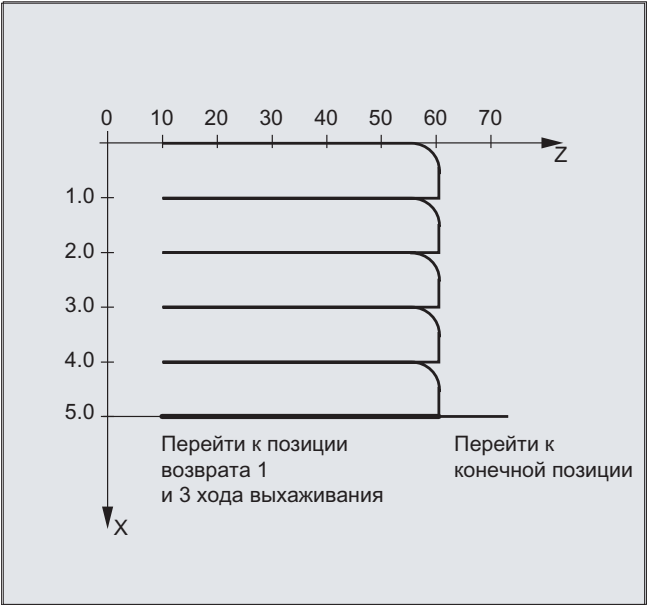
OSCILL:	Согласовать ось(и) подачи с качающейся осью
POSP:	Определить общую и частичные подачи (см. главу "Управление файлами и программами")
Конечная позиция:	Конечная позиция для оси подачи после прохождения всех частичных подач.
Длина фрагмента:	Величина частичной подачи в точке возврата/области возврата
Режим:	Разделение общей подачи на частичные подачи = два заключительных шага одинакового размера (предустановка); = все частичные подачи имеют одинаковый размер

Синхронные действия движения

WHEN... .. DO	если..., тогда...
WHENEVER ... DO	всегда когда..., тогда...

Пример

В точке возврата 1 подача не должна выполняться. Для точки возврата 2 подача должна быть выполнена уже на расстоянии ii2 перед точкой возврата 2 и качающаяся ось в точке возврата не должна ожидать завершения частичной подачи. Ось Z это качающаяся ось, а ось X – это ось подачи.



1. Параметры качания

Программный код	Комментарий
DEF INT ii2	; определить переменную для области возврата 2
OSP1[Z]=10 OSP2[Z]=60	; определить точку возврата 1 и 2
OST1[Z]=0 OST2[Z]=0	; точка возврата 1: точный останов точный Точка возврата 2: точный останов точный
FA[Z]=150 FA[X]=0.5	; подача качающейся оси Z, подача оси подачи X
OSCTRL[Z]=(2+8+16,1)	; отключить маятниковое движение в точке возврата 2; после RWL выхаживание и подвод к конечной позиции; после RWL подвод к соответствующей позиции возврата
OSNC[Z]=3	; ходы выхаживания
OSE[Z]=70	; конечная позиция = 70
ii2=2	; настроить область возврата
WAITP(Z)	; разрешить качание для оси Z

2. Синхронное действие движения

Программный код	Комментарий
WHENEVER \$AA_IM[Z]<\$SA_OSCILL_REVERSE_POS2[Z] DO -> \$AA_OVR[X]=0 \$AC_MARKER[0]=0	; Всегда, когда текущая позиция качающейся оси Z в MCS меньше, чем начало области возврата 2, тогда установить осевую процентровку оси подачи X на 0% и маркер с индексом 0 на значение 0.

15.2 Управляемое через синхронные действия качание (OSCILL)

Программный код	Комментарий
WHENEVER \$AA_IM[Z]>=\$SA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[Z]=0	; Всегда, когда текущая позиция качающейся оси Z в MCS больше и равно позиции возврата 2, тогда установить осевую процентовку качающейся оси Z 0%.
WHENEVER \$AA_DTEPW[X]==0 DO \$AC_MARKER[0]=1	; Всегда, когда остаточный путь частичной подачи равен 0, тогда установить меркер с индексом 0 на значение 1.
WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	; Всегда, когда меркер с индексом 0 равен 1, тогда установить осевую процентовку оси подачи X на 0%. Тем самым не допускается слишком ранняя подача (качающаяся ось Z еще не вышла из области возврата 2, но ось подачи X готова для новой подачи). Установить осевую процентовку качающейся оси Z с 0% (операция 2-го синхронного действия) для перемещения снова на 100%.

-> должно быть запрограммировано в одном кадре

3. Запустить качание

Программный код	Комментарий
OSCILL[Z]=(X) POSP[X]=(5,1,1)	; запуск осей Качающейся оси Z в качестве оси подачи присваивается ось X. Ось X должна двигаться до конечной позиции 5 с шагом в 1.
M30	; конец программы

Дополнительная информация

1. Определение параметров качания

Перед кадром движения, содержащим согласование оси подачи и качающейся оси, а также определение подачи, необходимо определить параметры для качания (см. "Асинхронное качание").

2. Определение синхронных действий движения

Через синхронные условия осуществляется:

Подавление подачи, пока качающаяся ось находится внутри области возврата (ii1, ii2) или на точке возврата (U1, U2).

Остановить маятниковое движение при подаче в точке возврата.

Снова **запустить маятниковое движение** после завершения частичной подачи.

Определить **старт следующей частичной подачи**.

3. Согласовать качающуюся ось и ось подачи, а также определить общую и частичную подачу.

Определение параметров качания

Согласование качающейся оси и оси подачи: OSCILL

OSCILL [<качающаяся ось>]=(<ось подачи1>,<ось подачи2>,<ось подачи3>)

15.2 Управляемое через синхронные действия качание (OSCILL)

С помощью команды "OSCILL" выполняются согласования осей и старт маятниковое движение.

Одной качающейся оси может быть присвоено макс. 3 оси подачи.

Примечание

Перед стартом качания необходимо определить синхронные условия для поведения осей.

Определение подач: POSP

POSP [<ось подачи>]=(<конечная позиция>,<длина фрагмента>,<режим>)

С помощью команды "POSP" СЧПУ сообщается:

- Общая подача (через конечную позицию)
- Размер соответствующей частичной подачи на точке возврата или в области возврата
- Поведение частичной подачи при достижении конечной позиции (через режим)

Режим = 0	Для обоих последних частичных подач выполняется разделение оставшегося пути до заданной точки на 2 заключительных шага равного размера (предусмотрено).
Режим = 1	Все частичные подачи имеют одинаковый размер. Они вычисляются из общей подачи.

Определение синхронных действий движения

Выполняемые в дальнейшем синхронные действия движения находят общее применение для качания.

Они находят примеры для решения отдельных задач, служащих элементами для создания прикладных пользовательских маятниковых движений.

Примечание

В отдельных случаях синхронные условия могут программироваться и по другому.

кодовых слов

WHEN ... DO ...	если..., тогда...
WHENEVER ... DO	всегда когда..., тогда...

Функции

С помощью подробно описанных ниже языковых средств могут быть реализованы следующие функции:

1. Подача в точке возврата.
2. Подача в области возврата.

3. Подача в обоих точках возврата.
4. Остановка маятникового движения в точке возврата.
5. Перезапуск маятникового движения.
6. Не запускать частичную подачу слишком рано.

Для всех представленных здесь в качестве примера синхронных действий имеются допущения:

- точка возврата 1 < точки возврата 2
- Z = качающаяся ось
- X = ось подачи

Примечание

Подробные пояснения см. главу "Синхронные действия движения".

Согласовать качающуюся ось и ось подачи, а также определить общую и частичную подачу

Подача в области возврата

Движение подачи должно начинаться в пределах области возврата до достижения точки возврата.

Эти синхронные действия препятствуют движению подачи до тех пор, пока качающаяся ось находится в области возврата.

С учетом приведенных выше допущений получаются следующие операторы:

Область возврата 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X] = 0
```

Всегда, когда актуальная позиция качающейся оси в MCS больше, чем начало области возврата 1, тогда установить осевую процентовку оси подачи на 0%.

Область возврата 2:

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X] = 0
```

Всегда, когда актуальная позиция качающейся оси в KS меньше, чем начало области возврата 2, тогда установить осевую процентовку оси подачи на 0%.

Подача в точке возврата

Пока качающаяся ось не достигла точки возврата, движение оси подачи не выполняется.

С учетом приведенных выше допущений получают следующие операторы:

Область возврата 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Всегда, когда актуальная позиция качающейся оси Z в MCS больше или меньше, чем позиция точки возврата 1, тогда установить осевую процентовку оси подачи X на 0% и осевую процентовку качающейся оси Z на 100%.

Область возврата 2:

Для точки возврата 2:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Всегда, когда актуальная позиция качающейся оси Z в MCS больше или меньше, чем позиция точки возврата 2, тогда установить осевую процентовку оси подачи X на 0% и осевую процентовку качающейся оси Z на 100%.

Остановка маятникового движения в точке возврата

Качающаяся ось останавливается на точке возврата, одновременно начинается движение подачи. Маятниковое движение продолжается после полного завершения движения подачи.

Одновременно это синхронное действие может использоваться для того, чтобы запустить движение подачи, если оно было остановлено предшествующим синхронным действием, которое еще активно.

С учетом приведенных выше допущений получают следующие операторы:

Область возврата 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Всегда, когда актуальная позиция качающейся оси в MCS равна позиции возврата 1, тогда установить осевую процентовку качающейся оси на 0% и осевую процентовку оси подачи на 100%.

Область возврата 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Всегда, когда актуальная позиция качающейся оси Z в MCS равна позиции возврата 2, тогда установить осевую процентовку качающейся оси X на 0% и осевую процентовку оси подачи на 100%.

Обработка точки возврата в режиме Online

Если на правой стороне сравнения стоит обозначенная с \$\$ переменная главного хода, то выполняется текущая обработка обеих переменных в такте IPO и их сравнение друг с другом.

Примечание

Подробности см. в главе "Синхронные действия движения".

Перезапуск маятникового движения

Это синхронное действие используется для того, чтобы продолжить движение качающейся оси, если движение частичной подачи завершено.

С учетом приведенных выше допущений получаются следующие операторы:

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z] = 100
```

Всегда, когда остаточный путь для частичной подачи оси подачи X в WCS равен нулю, тогда установить осевую процентовку качающейся оси на 100%.

Следующая частичная подача

После выполненной подачи необходимо не допустить преждевременного запуска следующей частичной подачи.

Для этого используется спец. для канала маркер (\$AC_MARKER[индекс]), который устанавливается в конце частичной подачи (частичный остаточный путь $\equiv 0$) и удаляется при выходе из области возврата. После синхронное действие не допускает следующего движения подачи.

С учетом приведенных выше допущений, к примеру, для точки возврата 1 получаются следующие операторы:

1. Установить маркер:

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[1]=1
```

Всегда, когда остаточный путь для частичной подачи оси подачи X в WCS равен нулю, тогда установить маркер с индексом 1 на 1.

2. Удалить маркер

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1] = 0
```

Всегда, когда актуальная позиция качающейся оси Z в MCS больше или меньше, чем позиция точки возврата 1, тогда установить маркер 1 на 0.

3. Недопущение подачи

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

Всегда, когда маркер 1 равен, тогда установить осевую процентовку оси подачи X на 0%.

Штамповка и вырубка

16.1 Активация/деактивация

16.1.1 Включение или выключение штамповки и вырубки (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

Активация/деактивация штамповки или вырубки

С PON и SON активируется функция штамповки или вырубки. SPOF завершает все спец. для штамповки и вырубки функции. Действующие модально команды PON и SON являются взаимоисключающими, т. е. PON деактивирует SON и наоборот.

Штамповка/вырубка с предварительным натяжением

Функции SONS и PONS также включают функции штамповки или вырубки.

В отличие от действующего при SON/PON управления ходом на уровне интерполяции в случае этих функций осуществляется сигнально-техническое управление запуском хода на сервоуровне. Благодаря этому работа может выполняться с увеличенной частотой хода и тем самым с увеличенной производительностью штамповки.

При обработке сигнала на предварительном натяжении все функции, ведущие к изменению позиции осей вырубки или штамповки (например, перемещение маховичком вручную, изменения фреймов через PLC, функции измерения) заблокированы.

Штамповка с задержкой

PDELAYON вызывает вывод хода штамповки с задержкой. Действующая модально команда имеет подготовленную функцию и тем самым как правило стоит перед PON. После PDELAYOF продолжается обычная штамповка.

Примечание

Время задержки устанавливается в установочных данных SD42400 \$SC_PUNCH_DWELLTIME.

Зависимое от хода ускорение

С PUNCHACC может быть определена характеристика ускорения, которая, в зависимости от расстояния между отверстиями, определяет различные ускорения.

Второй интерфейс штамповки

Станки, которые попеременно должны использовать второй интерфейс штамповки (второй перфоратор или схожее устройство), могут быть переключены на вторую пару быстрых цифровых входов и выходов СЧПУ (пара I/O). Для обоих интерфейсов штамповки доступна вся функциональность штамповки/вырубки. Переключение между

первым и вторым интерфейсом штамповки осуществляется через команды SPIF1 и SPIF2.

Примечание

Условие: Через машинные данные вторая пара I/O должна быть определена для функциональности штамповки (→ См. указания изготовителя станка!).

Синтаксис

```
PON G... X... Y... Z...
SON G... X... Y... Z...
SONS G... X... Y... Z...
PONS G... X... Y... Z...
PDELAYON
PDELAYOF
PUNCHACC (<Sмин>, <Амин>, <Sмакс>, <Амакс>)
SPIF1/SPIF2
SPOF
```

Объяснение

PON:	Активировать штамповку
SON:	Активировать вырубку
PONS:	Активировать штамповку с предварительным натяжением
SONS:	Активировать вырубку с предварительным натяжением
SPOF:	Деактивировать штамповку/вырубку
PDELAYON:	Активировать штамповку с задержкой
PDELAYOF:	Деактивировать штамповку с задержкой
PUNCHACC:	Активировать зависимое от хода ускорение
	Параметры:
	<Sмин> Мин. расстояние между отверстиями
	<Амин> Начальное ускорение <Амин> может быть больше, чем <Амакс> .
	<Sмакс> Макс. расстояние между отверстиями
SPIF1:	Активировать первый интерфейс штамповки
	Управление ходом выполняется через первую пару быстрых I/O.
SPIF2:	Активировать второй интерфейс штамповки
	Управление ходом выполняется через вторую пару быстрых I/O.
	Примечание: После RESET или запуска СЧПУ всегда активен первый интерфейс штамповки. Если используется только один интерфейс штамповки, то его программирование не требуется.

Примеры

Пример 1: Активировать вырубку

Программный код	Комментарий
...	
N70 X50 SPOF	; Позиционирование без запуска штамповки.
N80 X100 SON	; Активировать вырубку, запуск хода перед движением (X=50) и в конце запрограммированного движения (X=100) .
...	

Пример 2: штамповка с задержкой

Программный код	Комментарий
...	
N170 PDELAYON X100 SPOF	; Позиционирование без запуска штамповки, активация запуска штамповки с задержкой.
N180 X800 PON	; Активировать штамповку. После достижения конечной позиции ход штамповки выводится с задержкой.
N190 PDELAYOF X700	; Деактивировать штамповку с задержкой, обычный запуск высечки в конце запрограммированного движения.
...	

Пример 3: Штамповка с двумя интерфейсами штамповки

Программный код	Комментарий
...	
N170 SPIF1 X100 PON	; На конце кадра выполняется запуск хода на первом быстром выходе. Сигнал "Ход активен" контролируется на первом входе.
N180 X800 SPIF2	; Второй запуск хода выполняется на втором быстром выходе. Сигнал "Ход активен" контролируется на втором входе.
N190 SPIF1 X700	; Управление ходом для всех дальнейших ходов выполняется с первым интерфейсом.
...	

Дополнительная информация

Штамповка и вырубка с предварительным натяжением (PONS/SONS)

Штамповка и вырубка с предварительным натяжением невозможна в нескольких каналах одновременно. PONS или SONS может быть активирована только в одном канале соответственно.

Зависимое от хода ускорение (PUNCHACC)

Пример:

PUNCHACC (2, 50, 10, 100)

Расстояния между отверстиями меньше 2мм:

Движение выполняется с ускорением в 50% от макс. ускорения.

Расстояния между отверстиями от 2мм до 10мм:

Ускорение увеличивается пропорционально расстоянию до 100%.

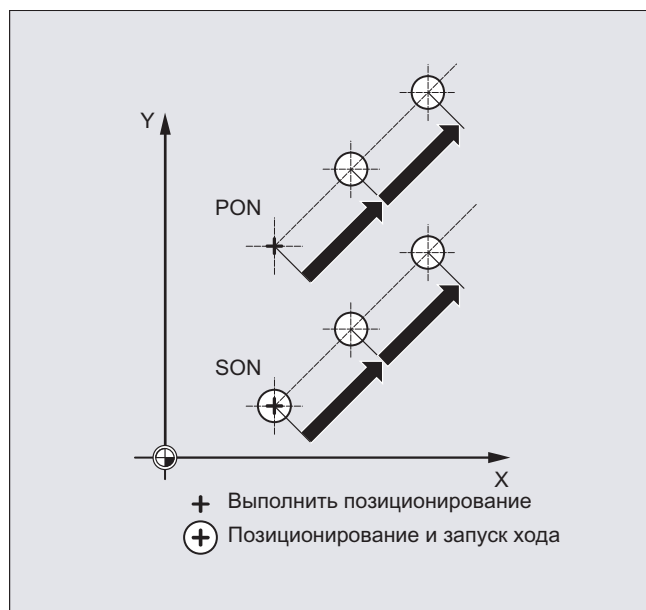
Расстояния между отверстиями больше, чем 10мм:

Движение с ускорением в 100%.

Запуск первого хода

Запуск первого хода после активизации функции осуществляется для вырубки и штамповки различно по времени:

- PON/PONS:
 - все ходы – включая ход первого кадра после активизации – осуществляются в конце кадра.
- SON/SONS:
 - первый ход после активизации вырубки осуществляется уже в начале кадра.
 - все последующие ходы соответственно запускаются в конце кадра.



Штамповка и вырубка на месте

Запуск хода осуществляется только тогда, когда кадр содержит информацию перемещения для осей штамповки или вырубки (оси активной плоскости).

Для того, чтобы все же запустить ход на том же месте, необходимо запрограммировать одну из осей штамповки/вырубки с путем перемещения 0.

Работа с вращающимися инструментами

Примечание

Для тангенциального подвода вращающихся инструментов к запрограммированной траектории необходимо использовать тангенциальное управление.

Использование команд М

С помощью техники макросов, как и прежде, можно использовать специальные функции М вместо языковых команд (совместимость). При этом действуют следующие соответствия с более старыми системами:

M20, M23	≡	SPOF
M22	≡	SON
M25	≡	PON
M26	≡	PDELAYON

Пример для файла макрокоманд:

Программный код	Комментарий
DEFINE M25 AS PON	; Штамповка вкл
DEFINE M125 AS PONS	; Штамповка с предварительным натяжением вкл
DEFINE M22 AS SON	; Вырубка вкл
DEFINE M122 AS SONS	; Вырубка с предварительным натяжением вкл
DEFINE M26 AS PDELAYON	; Штамповка с задержкой вкл
DEFINE M20 AS SPOF	; Штамповка, вырубка выкл
DEFINE M23 AS SPOF	; Штамповка, вырубка выкл

Пример программирования:

Программный код	Комментарий
...	
N100 X100 M20	; Позиционирование без запуска штамповки.
N110 X120 M22	; Активировать вырубку, запуск хода до и после движения.
N120 X150 Y150 M25	; Активировать штамповку, запуск хода в конце движения.
...	

16.2 Автоматическое разделение пути

Разделение на участки пути

При активной штамповке или вырубке как SPP, так и SPN приводит к разбивке всего запрограммированного для траекторных осей участка перемещения на определенное число участков пути одинаковой длины (эквидистантное разделение пути). Внутренне каждый участок перемещения соответствует одному кадру.

Число ходов

При штамповке первый ход осуществляется в конечной точке первого участка пути, при вырубке, напротив, в стартовой точке первого участка пути. Поэтому по всему участку перемещения получаются следующие цифры:

Штамповка: число ходов = число участков пути

Вырубка: число ходов = число участков пути + 1

Вспомогательные функции

Вспомогательные функции выполняются в первом из созданных кадров.

Синтаксис

SPP=

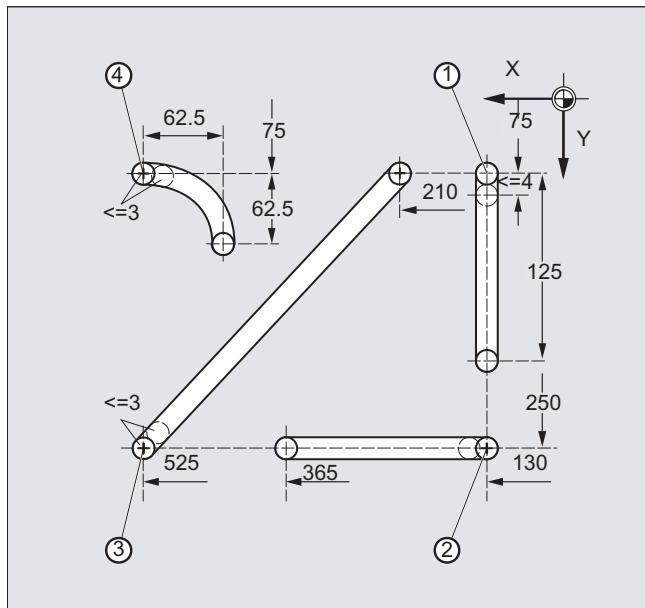
SPN=

Объяснение

SPP:	Размер участка пути (макс. интервал хода); действует модально
SPN:	Число участков пути на кадр; действует покадрово

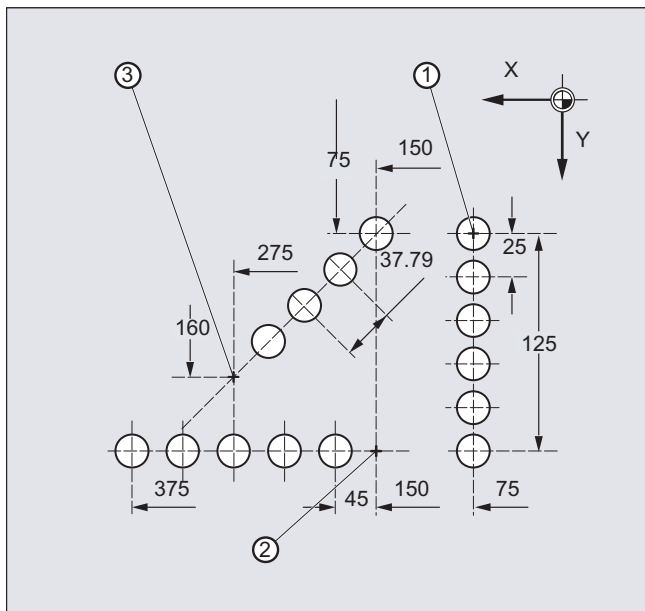
Пример 1

Запрограммированные участки вырубki должны быть автоматически разбиты на участки пути одинакового размера.



Программный код	Комментарий
N100 G90 X130 Y75 F60 SPOF	; Позиционирование на стартовую точку 1
N110 G91 Y125 SPP=4 SON	; Вырубка вкл; макс. длина участка пути для автоматического разделения пути: 4 мм
N120 G90 Y250 SPOF	; Вырубка выкл; позиционирование на стартовую точку 2
N130 X365 SON	; Вырубка вкл; макс. длина участка пути для автоматического разделения пути: 4 мм
N140 X525 SPOF	; Вырубка выкл; позиционирование на стартовую точку 3
N150 X210 Y75 SPP=3 SON	; Вырубка вкл; макс. длина участка пути для автоматического разделения пути: 3 мм
N160 X525 SPOF	; Вырубка выкл; позиционирование на стартовую точку 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; Вырубка вкл; макс. длина участка пути для автоматического разделения пути: 3 мм
N180 G00 G90 Y300 SPOF	; Вырубка выкл

Для отдельных рядов отверстий должно быть выполнено автоматическое разделение пути. Для разделения указывается макс. длина участка пути (значение SPP) соответственно.



Программный код	Комментарий
N100 G90 X75 Y75 F60 PON	; Позиционирование на стартовую точку 1; штамповка, штамповать отдельное отверстие
N110 G91 Y125 SPP=25	; Макс. длина участков пути для автоматического разделения пути: 25 мм
N120 G90 X150 SPOF	; Вырубка выкл; позиционирование на стартовую точку 2
N130 X375 SPP=45 PON	; Вырубка вкл; макс. длина участка пути для автоматического разделения пути: 45 мм
N140 X275 Y160 SPOF	; Вырубка выкл; позиционирование на стартовую точку 3
N150 X150 Y75 SPP=40 PON	; Штамповка вкл; вместо запрограммированной длины участков пути в 40 мм используется ;вычисленная длина участков пути в 37,79 мм.
N160 G00 Y300 SPOF	; Штамповка выкл; позиционирование

Длина участка пути SPP

С помощью SPP задается макс. интервал хода и тем самым макс. длина участков перемещения, на которые должен быть разделен весь участок перемещения. Отключение команды осуществляется со SPOF или SPP=0.

Пример:

N10 SON X0 Y0

N20 SPP=2 X10

Весь участок перемещения в 10 мм делится на 5 участков перемещения по 2 мм каждый (SPP=2).

Примечание

Разделение пути с SPP всегда выполняется эквидистантно: все участки пути имеют одинаковую длину. Это означает, что запрограммированное значение участков пути (значение SPP) действительно только тогда, когда коэффициент из всего участка перемещения и значения SPP является целочисленным. Если это не так, то размер участков пути уменьшается таким образом, чтобы получился целочисленный коэффициент.



Пример:

N10 G1 G91 SON X10 Y10

N20 SPP=3.5 X15 Y15

При всем участке перемещения в 15 мм и длине участков пути в 3,5 мм получается не целочисленный коэффициент (4.28). Поэтому выполняется уменьшение значения SPP до следующего возможного целочисленного коэффициента. В этом случае получается длина участков пути в 3 мм.

Число участков пути SPN

С помощью `SPN` определяется число участков пути, которые должны быть созданы из всего пути перемещения. Длина участков пути вычисляется автоматически. Так как `SPN` действует покадрово, то сначала необходимо активировать штамповку или вырубку с `PON` или `SON`.

16.2.2 Разделение пути для отдельных осей

Если наряду с траекторными осями и отдельные оси определены как ось штамповки-вырубки, то и они могут быть подвергнуты автоматическому разделению пути.

Поведение отдельной оси при SPP

Запрограммированная длина участка пути (`SPP`) всегда относится к траекторным осям. Поэтому в кадре, в котором наряду с движением отдельной оси и значением `SPP` траекторная ось не запрограммирована, значение `SPP` игнорируется.

Если в кадре запрограммирована и отдельная, и траекторная ось, то поведение отдельной оси зависит от настройки соответствующих машинных данных.

1. Стандартная настройка

Путь отдельной оси равномерно распределяется на созданные через `SPP` промежуточные кадры.

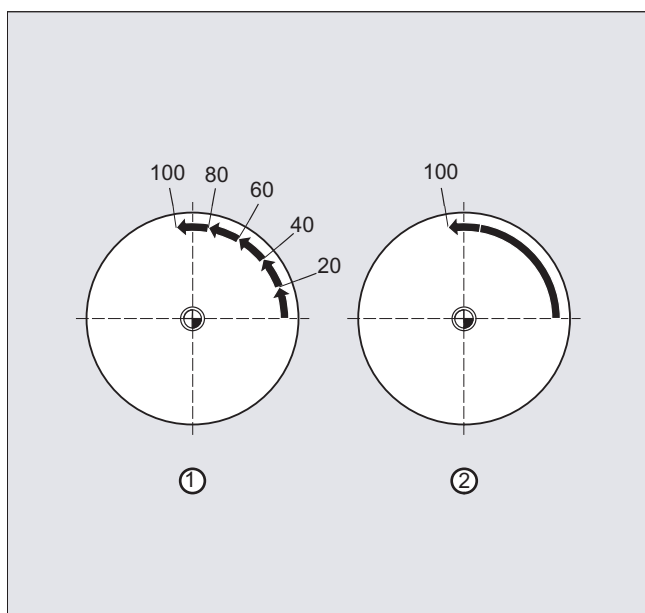
Пример:

```
N10 G1 SON X10 A0
```

```
N20 SPP=3 X25 A100
```

Посредством участка хода в 3 мм при всем участке перемещения оси X (траекторная ось) в 15 мм создается 5 кадров.

При этом ось A в каждом кадре поворачивается на 20 °.



1. Отдельная ось без разделения пути
Отдельная ось проходит весь свой путь в первом из созданных кадров.
2. Различное разделение пути
Поведение отдельной оси зависит от интерполяции траекторных осей:
 - Круговая интерполяция: Разделение пути
 - линейная интерполяция: нет разделения пути

Поведение при SPN

Запрограммированное число участков пути действует и в случае, если траекторная ось одновременно не запрограммирована.

Условие: Отдельная ось определена как ось штамповки/вырубki.

Шлифование

17.1 Включение / выключение специфического для шлифования контроля инструмента (TMON, TMOF)

Предварительно определенные процедуры TMON(...) и TMOF(...) включают / выключают специфичный для шлифования контроль инструмента (контроль геометрии и скорости).

Условие

Должны быть установлены специфичные для инструмента параметры инструмента \$TC_TPG1 - \$TC_TPG9.

Синтаксис

```
TMON (<Tnr>)
...
TMOF (<Tnr>)
```

Объяснение

TMON (. . .) :	Включение специфического для шлифования контроля инструмента Команда должна быть запрограммирована в канале, в котором должен быть включен специфический для шлифования контроль инструмента.
TMOF (. . .) :	Выключение специфического для шлифования контроля инструмента Команда должна быть запрограммирована в канале, в котором должен быть выключен специфический для шлифования контроль инструмента.
<Tnr>:	номер T Примечание: требуется только в том случае, если контроль вместо активного, находящегося в эксплуатации инструмента должен быть включен / выключен для неактивного шлифовального круга.
TMOF (0) :	Отключить контроль для всех инструментов

Другие функции

18.1 Осевые функции (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

"AXNAME" используется, к примеру, при создании общих циклов, когда имена осей неизвестны.

"AX" используется для косвенного программирования геометрических и синхронных осей. При этом идентификатор оси фиксируется в переменной типа AXIS или выводится командой, к примеру, "AXNAME" или "SPI".

"SPI" используется, когда осевые функции программируются для шпинделя, к примеру, синхронного шпинделя.

"AXTOSPI" используется для преобразования идентификатора оси в индекс шпинделя (обратная функция к "SPI").

"AXSTRING" используется для преобразования идентификатора оси (тип данных AXIS) в строку (обратная функция к "AXNAME").

"ISAXIS" используется в общих циклах, чтобы убедиться, что имеется определенная гео-ось и тем самым последующий вызов \$P_AXNX не будет отменен с ошибкой.

"MODAXVAL" для определения позиции модулю для круговых осей модулю.

Синтаксис

```
AXNAME ("строка")
AX [AXNAME ("строка")]
SPI (n)

AXTOSPI (A) или AXTOSPI (B) или AXTOSPI (C)
AXSTRING (SPI (n) )
ISAXIS (<номер гео-оси>)
<позиция модулю>=MODAXVAL (<ось>,<позиция оси>)
```

Объяснение

AXNAME:	Преобразует входную строку в идентификатор оси; входная строка должна содержать действительное имя оси.
AX:	Переменный идентификатор оси
SPI:	Преобразует номер шпинделя в идентификатор оси; передаваемый параметр должен содержать действительный номер шпинделя.
n:	Номер шпинделя
AXTOSPI:	Преобразует идентификатор оси в индекс шпинделя типа Integer. "AXTOSPI" соответствует обратной функции к "SPI".
X, Y, Z:	Идентификатор оси типа AXIS как переменная или постоянная
AXSTRING:	Выводится строка с согласованным номером шпинделя.

ISAXIS:	Проверяет наличие указанной гео-оси.
MODAXVAL:	Определяет для круговых осей модулю позицию модулю; она соответствует остатку модулю относительно спараметрированного диапазона модулю (составляет в стандартной установке 0 до 360 градусов; через MD30340 MODULO_RANGE_START и MD30330 \$MA_MODULO_RANGE можно изменить начало и размер диапазона модулю).

Примечание**Расширения SPI**

Осевая функция SPI(n) может использоваться и для чтения и записи компонентов фрейма. С ее помощью могут записываться фреймы, к примеру, с синтаксисом, \$P_PFRAME[SPI(1),TR]=2.22.

Посредством дополнительного программирования позиции оси через адрес AX[SPI(1)] = <позиция оси> ось может перемещаться. Условием этого является нахождение шпинделя в режиме позиционирования или осевом режиме.

Примеры**Пример 1: AXNAME, AX, ISAXIS**

Программный код	Комментарий
OVRA[AXNAME("поперечная ось")]=10	; Процентовка для поперечной оси
AX[AXNAME("поперечная ось")]=50.2	; Конечная позиция для поперечной оси
OVRA[SPI(1)]=70	; Процентовка для шпинделя 1
AX[SPI(1)]=180	; Конечная позиция для шпинделя 1
IF ISAXIS(1)==FALSE GOTOF WEITER	; Имеется ли абсцисса?
AX[\$P_AXN1]=100	; Перемещение по абсциссе
WEITER:	

Пример 2: AXSTRING

При программировании с AXSTRING[SPI(n)] более в качестве номера шпинделя не выводится индекс оси, с которой согласован шпиндель, а выводится строка "Sn".

Программный код	Комментарий
AXSTRING[SPI(2)]	; Выводится строка "S2".

Пример 3: MODAXVAL

Необходимо определить позицию модулю круговой оси модулю А.

Исходным значением для вычисления является позиция оси 372.55.

Спараметрированный диапазон модулю составляет 0 до 360 градусов:

MD30340 MODULO_RANGE_START = 0

MD30330 \$MA_MODULO_RANGE = 360

Программный код	Комментарий
R10=MODAXVAL (A, 372.55)	; Вычисленная позиция модуло R10 = 12.55.

Пример 4: MODAXVAL

Если запрограммированный идентификатор оси относится не к круговой оси модуло, то преобразуемое значение (<позиция оси>) возвращается без изменений.

Программный код	Комментарий
R11=MODAXVAL (X, 372.55)	; X это линейная ось; R11 = 372.55.

18.2 Переключаемые геометрические оси (GEOAX)

С помощью функции "Переключаемые гео-оси" можно заменять геометрические оси, сконфигурированные через машинные данные, на оси других каналов.

Синтаксис

GEOAX (<n>, <ось канала>, <n>, <ось канала>, <n>, <ось канала>)

GEOAX ()

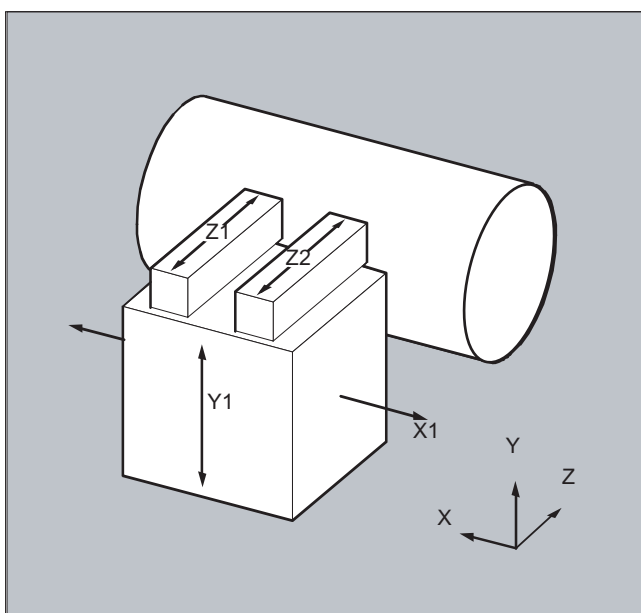
Объяснение

GEOAX (. . .)	Функция для переключения геометрических осей. Примечание: GEOAX () без указания параметра повторно активирует базовую конфигурацию гео-осей, спараметрированную в машинных данных.
<n>	Номер геометрической оси, которая заменяется на указанную ось канала. Диапазон значений: 0, 1, 2, 3 Примечание: 0: Указанная ось канала без замены удаляется из соединения геометрических осей 1: 1-я геометрическая ось $\hat{=}$ ось координат X (абсцисса) системы WCS 2: 2-я геометрическая ось $\hat{=}$ ось координат Y (ордината) системы WCS 3: 3-я геометрическая ось $\hat{=}$ ось координат Z (аппликата) системы WCS
<ось канала>	Имя оси канала, которая должна быть включена в соединение гео-осей.

Примеры

Пример 1: Попеременное включение двух осей как гео-оси

Инструментальная каретка может перемещаться через оси канала X1, Y1, Z1, Z2:



Гео-оси сконфигурированы таким образом, что после включения сначала Z1 действует как 3-ья гео-ось под именем гео-оси "Z" и образует вместе с X1 и Y1 соединение гео-осей.

В программе обработки детали оси Z1 и Z2 теперь должны попеременно работать в качестве гео-оси Z:

Программный код	Комментарий
...	
N100 GEOAX(3,Z2)	; Как 3-я гео-ось (Z) используется ось канала Z2.
N110 G1 ...	
N120 GEOAX(3,Z1)	; Как 3-я гео-ось (Z) используется ось канала Z1.
...	

Пример 2: Переключение гео-осей при 6 осях канала

Станок имеет 6 осей канала с именами XX, YY, ZZ, U, V, W.

Первичная установка конфигурации гео-осей через машинные данные это:

Ось канала XX = 1-я гео-ось (ось X)

Ось канала YY = 2-я гео-ось (ось Y)

Ось канала ZZ = 3-я гео-ось (ось Z)

Программный код	Комментарий
N10 GEOAX()	; Базовая конфигурация гео-осей активна.
N20 G0 X0 Y0 Z0 U0 V0 W0	; Все оси ускоренным ходом на позицию 0.
N30 GEOAX(1,U,2,V,3,W)	; Ось канала U становится первой (X), V – второй (Y) ; и W – третьей гео-осью (Z).
N40 GEOAX(1,XX,3,ZZ)	; Ось канала XX становится первой (X), ZZ – третьей ; гео-осью (Z). Ось канала V остается второй ; гео-осью (Y).

Программный код	Комментарий
N50 G17 G2 X20 I10 F1000	; Полный круг в плоскости X/Y. Двигаются ; оси канала XX и V.
N60 GEOAX(2,W)	; Ось канала W становится второй гео-осью (Y) .
N80 G17 G2 X20 I10 F1000	; Полный круг в плоскости X/Y. Двигаются ; оси канала XX и W.
N90 GEOAX()	; Сброс на исходное состояние.
N100 GEOAX(1,U,2,V,3,W)	; Ось канала U становится первой (X) , V – второй ; (Y) и W – третьей гео-осью (Z) .
N110 G1 X10 Y10 Z10 XX=25	; Оси канала U, V, W движутся на ; позицию 10. XX как дополнительная ось движется на позицию 25.
N120 GEOAX(0,V)	; V изымается из соединения гео-осей. ; U и W остаются первой (X) и третьей ; гео-осями (Z) . ; Вторая гео-ось (Y) остается свободной.
N130 GEOAX(1,U,2,V,3,W)	; Ось канала U остается первой (X) , V становится ; второй (Y) , W остается третьей гео-осью (Z) .
N140 GEOAX(3,V)	; V становится третьей гео-осью (Z) , при этом W ; переписывается и тем самым изымается из ; соединения гео-осей. Вторая гео-ось (Y) , ; как и прежде, свободна.

Машинные данные

Конфигурация осей

Согласование геометрических осей, дополнительных осей и осей станка с осями каналов:

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB
- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB
- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB
- MD20070 \$MC_AXCONF_MACHAX_USED
- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB
- MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX

Поведение при Reset

Поведение измененных согласований геометрических осей при Reset:

- MD20110 \$MC_RESET_MODE_MASK, бит 12
- MD20118 \$MC_GEOAX_CHANGE_RESET

Поведение при запуске ЧПУ

- MD20112 \$MC_START_MODE_MASK, бит 12

Сообщение для программы электроавтоматики

Возможность параметрирования M-команды, которая при переключении геометрических осей выдается на интерфейс ЧПУ/PLC:

- MD22532 \$MC_GEOAX_CHANGE_M_CODE

Граничные условия**Нет переключения геометрических осей**

- Если активна одна из следующих функций, переключение геометрических осей невозможно:
 - Трансформация
 - Сплайн-интерполяция
 - Коррекция на радиус инструмента
 - Точная коррекция на инструмент
- Геометрическая ось и дополнительная ось канала имеют одно и то же имя.
- Одна из осей, участвующих в переключении геометрических осей, участвует в действии, которое продолжается за границы кадра. Например, выходящая за границы кадра позиционирующая или ведомая ось соединения осей.

Круговые оси

Круговые оси нельзя сделать геометрическими осями.

Состояние оси после замены

Замененная через переключение в соединении гео-осей ось после процесса переключения через своё имя оси канала может быть запрограммирована как дополнительная ось.

Фреймы, защищенные области, ограничения рабочего поля

При переключении гео-осей все фреймы, защищенные области и ограничения рабочего поля удаляются.

Полярные координаты

Смена геометрических осей с помощью GEOAX аналогично смене плоскостей через G17-G19 устанавливает модальные полярные координаты на значение 0.

DRF, NPV

Возможное смещение маховичком (DRF) или внешнее смещение нулевой точки (WO) продолжает действовать после переключения.

Базовая конфигурация гео-осей

Команда GEOAX () вызывает базовую конфигурацию соединения геометрических осей.

После POWER ON и при переключении в режим работы "Реферирование" выполняется автоматический возврат к базовой конфигурации.

Коррекция длин инструмента

Активная коррекция длин инструмента действует и после процесса переключения. Но для новых принятых или изменившихся позиции гео-осей она считается еще не выведенной. При первой команде движения для этих гео-осей результирующий путь перемещения состоит соответственно из суммы коррекции длин инструмента и запрограммированного пути перемещения.

Гео-оси, сохранившие при переключении свою позицию в структуре осей, сохраняют и свой статус касательно коррекции длин инструмента.

Конфигурация гео-осей при активной трансформации

- Конфигурацию гео-осей, спараметрированную для активной трансформации через машинные данные трансформации, изменить с помощью функции "Переключаемые гео-оси" невозможно.
- Различная конфигурация гео-осей для трансформации, в машинных данных трансформации нужно спараметризовать разные блоки данных.
- Измененная через GEOAX конфигурация гео-осей при активации трансформации удаляется.
- В отношении гео-осей варианты их параметрирования для конкретных трансформаций при активных трансформациях имеют приоритет перед вариантами параметрирования, связанными с переключением гео-осей.
Пример: Одна из трансформаций активна. Согласно машинным данным эта трансформация при Reset канала должна сохраниться. Но одновременно при Reset канала должна восстановиться базовая конфигурация гео-осей. Сохраняется конфигурация гео-осей, которая задана для трансформации.
- С выключением трансформации спараметрированная первичная установка конфигурации гео-осей становится снова активной.

Режим работы JOG, функция станка REF

При переключении на режим работы JOG, функция станка REF (реферирование), спараметрированная в машинных данных конфигурация гео-осей становится активной.

18.3 Осевого контейнер (AXCTSWE, AXCTSWED, AXCTSWEC)

Через команды "AXCTSWE" или "AXCTSWED" разрешается вращение указанного осевого контейнера.

С помощью команды "AXCTSWEC" уже данное разрешение на вращение осевого контейнера может быть снова отменено.

Синтаксис

AXCTSWE (<ID>)
 AXCTSWED (<ID>)
 AXCTSWEC (<ID>)

Объяснение

AXCTSWE:	Разрешение на вращение осевого контейнера Обработка программы через "AXCTSWE" не останавливается. Вращение выполняется после того, как все участвующие в осевом контейнере каналы дали разрешение.	
AXCTSWED:	Разрешение на вращение осевого контейнера без учета других участвующих в контейнере каналов Указание • Вариант команды специально для упрощения ввода в эксплуатацию программы обработки детали или синхронного действия. • Поведение касательно других участвующих в осевом контейнере каналов может быть задано через: MD12760 \$MN_AXCT_FUNCTION_MASK, Bit 0	
AXCTSWEC:	Отмена разрешения на вращение осевого контейнера Указание Разрешение вращения осевого контейнера может быть отменено, только если вращение еще не началось: \$AN_AXCTSWA[<осевой контейнер>] == 0 По системным переменным см. "Осевого контейнер (AXCTSWE, AXCTSWED, AXCTSWEC) (Страница 679)"	
<ID>:	Идентификатор осевого контейнера или оси контейнера:	
	СТ<номер>:	Идентификатор по умолчанию осевого контейнера: MD12750 \$MN_AXCT_NAME_TAB Пример: "СТ1"
	<контейнер>:	Спец. для пользователя идентификатор осевого контейнера: MD12750 \$MN_AXCT_NAME_TAB Пример: "CONTAINER_1"
	<ось>:	Идентификатор известной в канале оси контейнера

Примечание**Размер шага**

Размер шага вращения осевого контейнера задается через установочные данные:

SD41700 \$SN_AXCT_SWWIDTH

Дополнительная информация**Диагностика**

Актуальное состояние осевого контейнера может быть считано через следующие системные переменные:

Системная переменная	Тип	Описание
\$AC_AXCTSWA[<имя>]	BOOL	Спец. для канала состояние осевого контейнера
\$AN_AXCTSWA[<осевой контейнер>]	BOOL	Спец. для NCU состояние осевого контейнера
\$AN_AXCTSWE[<осевой контейнер>]	INT	Спец. для слота состояние вращения осевого контейнера Системная переменная выводит по битам состояние слотов осевого контейнера. Каждый бит соответствует одному слоту.
\$AN_AXCTAS[<осевой контейнер>]	INT	Число мест (слотов), на которое выполнено текущее продвижение осевого контейнера.

Вращение осевого контейнера с не явной GET / GETD

Через следующие машинные данные можно установить, что при команде "AXCTSWE" все оси контейнера канала посредством не явной "GET / GETD" будут переданы в канал. Переход оси снова возможен только после завершения вращения контейнера.

MD10722 \$MN_AXCHANGE_MASK, бит 1 = 1

Примечание

Вращение осевого контейнера с не явной "GET / GETD" **не** выполняется для оси в состоянии "Ось главного хода" (к примеру, PLC-ось), т.к. ось для вращения осевого контейнера должна выйти из этого состояния.

18.4 Ожидание действительной позиции оси (WAITENC)

С помощью языковой команды "WAITENC" в программе ЧПУ можно подождать, пока для сконфигурированных с MD34800 \$MA_WAIT_ENC_VALID = 1 осей не будут доступны синхронизированные или восстановленные позиции осей.

В состоянии ожидания возможно прерывание, к примеру, через запуск ASUP или через смену режимов работы на JOG. При продолжении программы состояние ожидания при необходимости возобновляется.

Примечание

Состояние ожидания отображается на интерфейсе через режим приостановки "Ожидание измерительной системы".

Синтаксис

"WAITENC" может быть запрограммирована в программном блоке любой программы ЧПУ.

Программирование должно выполняться в отдельном кадре:

```
...
WAITENC
...
```

Пример

"WAITENC" используется, к примеру, в управляемой событием программе пользователя .../_N_CMA_DIR/_N_PROG_EVENT_SPF, как показывает следующий пример применения.

Пример использования: Отвод инструмента после POWER OFF с трансформацией ориентации

Обработка с ориентацией инструмента была отменена через прерывание подачи питания.

При последующем запуске вызывается управляемая событиями программа пользователя.../_N_CMA_DIR/_N_PROG_EVENT_SPF.

В управляемой событиями программе пользователя с "WAITENC" выполняется ожидание синхронизированных или восстановленных позиций осей, чтобы после можно было бы вычислить фрейм, который точно устанавливает WCS в направлении инструмента.

Программный код	Комментарий
...	
IF \$P_PROG_EVENT == 4	; Запуск.
IF \$P_TRAFO <> 0	; Трансформация была выбрана.
WAITENC	; Ожидать действительных осевых позиций осей ориентации.
TOROTZ	; Повернуть Z-ось WCS в направлении оси инструмента.
ENDIF	

18.4 Ожидание действительной позиции оси (WAITENC)

Программный код	Комментарий
-----------------	-------------

M17	
ENDIF	
...	

После этого возможен свободный ход инструмента в режиме работы JOG через движение отвода в направлении оси инструмента.

18.5 Программируемое переключение блоков параметров (SCPARA)

С помощью команды "SCPARA" можно запросить для оси переключение на определенный блок параметров.

Примечание

Без переключения блоков параметров при обработке резьбы

При резьбонарезании G33 и нарезании внутренней резьбы G331 / G332 блок параметров выбирается СЧПУ и не может быть изменен.

Заблокированное переключение блоков параметров

Переключение блоков параметров может быть запрошено и через интерфейс ЧПУ/PLC. Для недопущения конфликтов переключения, через интерфейс ЧПУ/PLC можно заблокировать переключение блоков параметров ЧПУ (SCPARA):

DB31, ... DBX9.3 (установка блока параметров через ЧПУ заблокирована)

Примечание

Если переключение блоков параметров запрашивается через "SCPARA", в то время как переключение блоков параметров через интерфейс ЧПУ/PLC заблокировано, то переключение отклоняется без сообщения об ошибке.

Синтаксис

SCPARA [<ось>]=<значение>

Значение

SCPARA:	Команда: Переключить блок параметров	
<ось>:	Идентификатор оси (ось канала)	
	Тип:	AXIS
<значение>:	Номер блока параметров: 1, 2, 3, ... макс. номер блока параметров	

Пример

Программный код	Комментарий
...	
N110 SCPARA[X] = 3	; Выбор: ось X, 3-й блок параметров
...	

Дополнительная информация

Разрешение переключения блоков параметров

Переключение блоков параметров оси должно быть разрешено явно:

18.5 Программируемое переключение блоков параметров (SCPARA)

MD35590 \$MA_PARAMSET_CHANGE_ENABLE[<ось>]

Чтение номера блока параметров

Номер выбранного блока параметров (заданный блок параметров) может быть считан через системную переменную \$AA_SCPAR.

Литература

Подробную информацию по блокам параметров можно найти в:

Описание функций - Основные функции; глава "Скорости, системы заданного/ фактического значения, регулирование (G2)" > "Регулирование" > "Блоки параметров регулятора положения"

18.6 Проверка имеющейся языковой среды ЧПУ (STRINGIS)

С помощью функции "STRINGIS(...)" можно проверить, доступна ли указанная строка как элемент языка программирования ЧПУ в текущей языковой среде.

Определение

```
INT STRINGIS (STRING <имя>)
```

Синтаксис

```
STRINGIS (<имя>)
```

Объяснение

STRINGIS:	Функция с возвращаемым значением
<имя>:	Имя проверяемого элемента языка программирования ЧПУ
Возвращаемое значение:	Формат возвращаемого значения ухх (дес.).

Элементы языка программирования ЧПУ

Могут быть проверены следующие элементы языка программирования ЧПУ:

- G-команды всех существующих групп G-функций, например, "G0", "INVCW", "POLY", "ROT", "KONT", "SOFT", "CUT2D", "CDON", "RMBBL", "SPATH"
- Адреса DIN или ЧПУ, например, "ADIS", "RNDM", "SPN", "SR", "MEAS"
- Функции, например, "TANG(...)" или "GETMDACT"
- Процедуры, например, "SBLOF".
- Кодовые слова, например, "ACN", "DEFINE" или "SETMS"
- Системные данные например, машинные данные \$M... , установочные данные \$S... или опциональные данные \$O...
- Системные переменные \$A... , \$V... , \$P...
- R-параметры R...
- Имена активированных циклов
- Переменные GUD и LUD
- Имена макрокоманд
- Имена меток

Возвращаемое значение

Возвращаемое значение релевантно только в первых 3 десятичных разрядах. Формат возвращаемого значения ухх, где у = базовая информация и хх = подробная информация.

Возвращаемое значение	Объяснение
000	Строка 'name' неизвестна в данной системе ¹⁾
100	Строка 'name' это элемент языка программирования ЧПУ, но сейчас не может быть запрограммирована (опция/функция не активна)
2хх	Строка 'name' это программируемый элемент языка программирования ЧПУ (опция/функция активна). Подробная информация хх содержит дополнительную информацию по типу элемента:
	хх Значение
	01 Адрес DIN или адрес ЧПУ ²⁾
	02 G-команда (например, G04, INVCW)
	03 Функция с возвращаемым значением
	04 Функция без возвращаемого значения
	05 Кодовое слово (например, DEFINE)
	06 Машинные (\$M...), установочные (\$S...) или опциональные данные (\$O...)
	07 Системные параметры, например, системная переменная (\$...) или R-параметр (R...)
	08 Цикл (цикл должен быть загружен в ЧПУ, программы циклов должны быть активны ³⁾)
	09 GUD-переменная (GUD-переменная должна быть определена в файле определений GUD и активирована)
	10 Имя макрокоманды (макрокоманда должна быть определена в файлах определения макрокоманд и активирована) ⁴⁾
	11 LUD-переменная текущей программы обработки детали
	12 ISO G-команда (языковой режим ISO должен быть активен)
400	Строка 'name' это адрес ЧПУ, который не был определен как хх == 01 или хх == 10 и не является G или R ²⁾
у00	Спец. согласование невозможно

1) В зависимости от СЧПУ при определенных обстоятельствах известно только подмножество языковых команд ЧПУ Siemens, например, SINUMERIK 802D sl. На этих СЧПУ для строк, которые в принципе являются языковыми командами ЧПУ Siemens, возвращается значение 0. Это поведение можно изменить через MD10711 \$MN_NC_LANGUAGE_CONFIGURATION. В этом случае при MD10711 = 1 для языковых команд ЧПУ Siemens всегда возвращается значение 100.

2) Адресами ЧПУ являются следующие буквы: A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z. Эти адреса ЧПУ могут быть запрограммированы и с расширением адреса. Расширение адреса может быть указано при проверке со STRINGIS. Пример: 201 == STRINGIS("A1").

Буквы: D, F, H, L, M, N, O, P, S, T это адреса ЧПУ или вспомогательные функции, которые определяются пользователем. Для них всегда возвращается значение 400. Пример: 400 == STRINGIS("D"). Эти адреса ЧПУ при проверке со STRINGIS не могут указываться с расширением адреса.

Пример: 000 == STRINGIS("M02"), но 400 == STRINGIS("M").

3) Имена параметров циклов невозможно проверить посредством STRINGIS.

4) Определенный как макрос адрес, например, G, H, M, L, идентифицируется как макрос.

Примеры

В следующих примерах допускается, что указанный в качестве строки языковой элемент ЧПУ, если не указано иначе, в принципе может быть запрограммирован в СЧПУ.

1. Строка "T" определена как вспомогательная функция:
400 == STRINGIS ("T")
000 == STRINGIS ("T3")
 2. Строка "X" определена как ось:
201 == STRINGIS ("X")
201 == STRINGIS ("X1")
 3. Строка "A2" определена как адрес ЧПУ с расширением:
201 == STRINGIS ("A")
201 == STRINGIS ("A2")
 4. Строка "INVCW" определена как упомянутая G-команда:
202 == STRINGIS ("INVCW")
 5. Строка "\$MC_GCODE_RESET_VALUES" определена как машинные данные:
206 == STRINGIS (" \$MC_GCODE_RESET_VALUES")
 6. Строка "GETMDACT" это языковая функция ЧПУ:
203 == STRINGIS ("GETMDACT ")
 7. Строка "DEFINE" это кодовое слово:
205 == STRINGIS ("DEFINE")
 8. Строка "\$TC_DP3" это системный параметр (компонент длин инструмента):
207 == STRINGIS (" \$TC_DP3")
 9. Строка "\$TC_TP4" это системный параметр (размер инструмента):
207 == STRINGIS (" \$TC_TP4")
 10. Строка "\$TC_MPP4" это системный параметр (состояние места в магазине):
 - Управление магазином инструментов активно: 207 ==
STRINGIS (" \$TC_MPP4") ;
 - Управление магазином инструментов не активно: 000 ==
STRINGIS (" \$TC_MPP4")
- См. также ниже абзац: управление магазином инструментов.
11. Строка "MACHINERY_NAME" определена как GUD-переменная:
209 == STRINGIS ("MACHINERY_NAME")
 12. Строка "LONGMACRO" определена как макрос:
210 == STRINGIS ("LONGMACRO")
 13. Строка "MYVAR" определена как LUD-переменная:
211 == STRINGIS ("MYVAR")
 14. Строка "XYZ" — неизвестная в ЧПУ команда, GUD-переменная, имя макрокоманды или цикла:
000 == STRINGIS ("XYZ")

Управление магазином инструментов

Если функция управления магазином инструментов не активна, то STRINGIS возвращает для системных параметров управления магазином инструментов, независимо от машинных данных

- MD10711 \$MN_NC_LANGUAGE_CONFIGURATION

всегда значение 000.

Режим ISO

Если функция "Режим ISO" активна:

- MD18800 \$MN_MM_EXTERN_LANGUAGE (активация внешних языков ЧПУ)
- MD10880 \$MN_MM_EXTERN_CNC_SYSTEM (адаптируемая система управления)

STRINGIS сначала проверяет указанную строку как G-команду SINUMERIK. Если строка не является G-командой SINUMERIK, то затем она проверяется как G-команда ISO.

Запрограммированные переключения (G290 (режим SINUMERIK), G291 (режим ISO)) не влияют на STRINGIS.

Пример

Релевантные для функции STRINGIS(...) машинные данные имеют следующие значения:

- MD10711 \$MN_NC_LANGUAGE_CONFIGURATION = 2 (как известные рассматриваются только такие языковые команды ЧПУ, опции которых установлены)
- MD19410 \$ON_TRAFO_TYPE_MASK = 'H0' (опция: трансформации)
- MD10700 \$MN_PREPROCESSING_LEVEL='H43' (предварительная обработка для циклов активна)

Следующая иллюстративная программа выполняется без сообщения об ошибке:

Программный код	Комментарий
N1 R1=STRINGIS("TRACYL")	; R1 == 0, т.к. TRACYL из-за отсутствующей опции трансформации определяется как "не известная"
N2 IF STRINGIS("TRACYL") == 204	
N3 TRACYL(1,2,3)	; N3 пропускается
N4 ELSE	
N5 G00	; и вместо этого выполняется N5
N6 ENDIF	
N7 M30	

18.7 Интерактивный вызов окон из программы обработки детали (MMC)

Посредством предопределенной подпрограммы MMC(...) из программы ЧПУ можно отобразить пользовательские диалоги на интерфейс пользователя, например, SINUMERIK Operate.

Возможно конфигурирование диалогов следующих типов:

- Run MyScreens
- Easy XML
- XML пользователя

Литература

- Руководство по программированию Run MyScreens
- Руководство по программированию Easy XML

Синтаксис

MMC ("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>" , "<QUIT>")

Объяснение

MMC (. . .) :	идентификатор подпрограммы Параметры кодируются по месту и разделяются запятой в пределах двух строк — командной строки и строки квитирования.	
Параметры внутри командной строки:		
<ADDRESS>:	Область управления, в которой выполняются сконфигурированные диалоги с пользователем.	
	Функция	Области управления
	Диалог с пользователем "Run MyScreens"	CYCLES
	Диалог с пользователем "Easy XML"	CYCLES
	XML пользователя	XML
	Всплывающее окно "Run MyScreens"	POPUPDLG
	Всплывающее окно "Easy XML"	POPUPDLG
<COMMAND>:	Выполняемая команда	
	Функция	Команды
	Диалог с пользователем "Run MyScreens"	PICTURE_ON, PICTURE_OFF
	Диалог с пользователем "Easy XML"	PICTURE_ON, PICTURE_OFF
	XML пользователя	XML_ON, XML_OFF
	Всплывающее окно "Run MyScreens"	PICTURE_ON, PICTURE_OFF
	Всплывающее окно "Easy XML"	PICTURE ON, PICTURE OFF

<FILE>:	Имя файла, в котором запрограммирован отображаемый диалог	
	Функция	Файлы
	Диалог с пользователем "Run MyScreens"	<name>.com
	Диалог с пользователем "Easy XML"	<name>.xml
	XML пользователя	<name>.xml
	Всплывающее окно "Run MyScreens"	<name>.com
	Всплывающее окно "Easy XML"	<name>.xml
<DIALOG>:	Имя отображаемого диалога	
	Функция	Имя диалога
	Все функции, кроме всплывающего окна "Easy XML" с конфигурированием непосредственно в программе ЧПУ	Имя диалога, сконфигурированное в файле <FILE>
	Всплывающее окно "Easy XML" с конфигурированием непосредственно в программе ЧПУ (см. пример 3)	main
Параметры внутри строки квитирования:		
<QUIT>:	Тип квитирования	
	N:	Нет квитирования Обработка программы продолжается после отправки команды. Квитирование не выполняется, если команду не удалось выполнить успешно. Указание Тип квитирования "N" используется, если в программе ЧПУ программируется время воспроизведения изображения (время ожидания) (см. ниже Пример 2)
	A:	Асинхронное квитирование Обработка программы продолжается после подачи команды. Возвращаемое значение сохраняется в спец. переменной квитирования пользователя, определенной в рамках конфигурирования диалога (GUD-переменной) и может считываться в программе ЧПУ.

Пример

Пример 1

Индикация диалога и реакция на пользовательское управление в программе ЧПУ.

Программный код	Комментарий
; Переменная квитирования QUIT уже была создана как глобальная переменная пользователя (GUD)	
; типом STRING при конфигурировании диалога:	
; DEF NCK STRING[20] QUIT	
QUIT = "XXX"	; Инициализация переменной квитирования
G4 F5	

18.7 Интерактивный вызов окон из программы обработки детали (MMC)

Программный код	Комментарий
MMC ("CYCLES, PICTURE_ON, test.com, test1", "A")	; Показать диалог ; — Область управления: CYCLES ; — Статус изображения: PICTURE_ON (отобразить) ; — Файл изображения диалога: test.com ; — Изображение диалога: test1
INPUT:	; Ожидание пользовательского управления
STOPRE	; Остановка предварительной обработки
IF MATCH (QUIT, "RUN") >= 0 GOTOF WORK	; Программная клавиша "RUN"
IF MATCH (QUIT, "CHK") >= 0 GOTOF CHECK	; Программная клавиша "CHK"
GOTOF INPUT	; => Ожидание
WORK:	; Программная клавиша "RUN" нажата
MSG ("Продолжить обработку -> Пуск ЧПУ")	; Вывести сообщение
MMC ("CYCLES, PICTURE_OFF", "N")	; Закрыть диалог
M0	; Ожидание пуска ЧПУ
GOTOF END	; => конец программы
CHECK:	; Программная клавиша "CHK" нажата
MSG ("Подвести к позиции -> Пуск ЧПУ")	; Вывести сообщение
MMC ("CYCLES, PICTURE_OFF", "N")	; Закрыть диалог
M0	; Ожидание пуска ЧПУ
GOTOF END	; => конец программы
END:	
...	

Пример 2

Время отображения диалога в программе ЧПУ устанавливается, например, через время ожидания.

Программный код	Комментарий
F1000 G94	
...	
MMC ("POPUPDLG, PICTURE_ON, xmldial_emb.xml, main", "N")	; Показать диалог
X200	
Z40	
MMC ("POPUPDLG, PICTURE_OFF", "N")	; Закрыть диалог

Пример 3

Введение скрипта всплывающего окна в программу ЧПУ и его использование.

Программный код

```
PROC POPUP_TEST
```

```
; ----- Скрипт -----
```

Программный код

```

; <main_dialog entry="rpara_main">
;   <let name="xpos" />
;   <let name="ypos" />
;   <let name="field_name" type="string" />
;   <let name="num" />
;   <menu name="rpara_main">
;     <open_form name="rpara_form"/>
;     <softkey_back>
;       <close_form />
;     </softkey_back>
;   </menu>
;   <form name="rpara_form">
;     <init>
;       <caption>mask from NC part program</caption>
;       <let name="count" >0</let>
;       <op>
;         xpos = 120;
;         ypos = 34;
;         "nck/Channel/Parameter/R[10]" = 10;
;       </op>
;       <!-- load the number of controls -->
;       <op>
;         num = "nck/Channel/Parameter/R[10]";
;       </op>
;       <while>
;         <condition> count < num</condition>
;         <print name="field_name" text="edit%d">count</print>
;         <op>
;           ypos = ypos + 24;
;           count = count + 1;
;         </op>
;       </while>
;     </init>
;     <paint>
;       <op>
;         xpos = 8;
;         ypos = 36;
;         count = 0;
;       </op>
;       <while>
;         <condition>count < num</condition>
;         <print name="field_name" text="R-Parameter%d">count</print>
;         <text xpos = "$xpos" ypos = "$ypos" >$$$field_name</text>
;       </op>

```

Программный код

```

;          ypos = ypos + 24;
;          count = count + 1;
;          </op>
;          </while>
;          </paint>
;        </form>
; </main_dialog>
; ===== Блок программы =====
...
G94 F100
MMC ("POPUPDLG, PICTURE_ON, xmldial_emb.xml, main", "N")
G4 F4
X200
MMC ("POPUPDLG, PICTURE_OFF", "N")
G4 F2
X0
...

```

Граничные условия

- Файлы определения *.com диалогов необходимо сохранять в папке "roj".
- Файлы определения Easy XML *.xml диалогов необходимо сохранять в папке "appl". Если файлы определений сохранены в другой директории, указание пути должно быть непрямым, исходя из директории "appl".
- Спец. диалоги пользователя невозможно отображать одновременно из различных каналов.

18.8 Время выполнения программы / счетчик деталей

Для поддержки оператора станка предоставляется информация по времени выполнения программы и числу деталей.

Эту информацию может обрабатывать как системные переменные в программе ЧПУ и/или PLC. Одновременно эта информация доступна для индикации на интерфейсе пользователя.

18.8.1 Время выполнения программы

Функция "Время выполнения программы" предоставляет внутренние таймеры ЧПУ для контроля технологических процессов, которые могут считываться через спец. для ЧПУ и канала системные переменные в программе обработки детали и в синхронных действиях.

Триггер для измерения времени выполнения (\$AC_PROG_NET_TIME_TRIGGER) это единственная записываемая системная переменная функции и служит для выборочного измерения сегментов программы. Т.е. через запись в триггер в программе ЧПУ измерение времени может включаться и снова выключаться.

Системная переменная	Объяснение	Активность
Спец. для ЧПУ		
\$AN_SETUP_TIME	Время от последнего запуска СЧПУ со стандартными значениями ("холодный пуск") в минутах При каждом запуске СЧПУ со стандартными значениями автоматически сбрасывается на "0".	• Всегда активен
\$AN_POWERON_TIME	Время от последнего обычного запуска СЧПУ ("горячий пуск" в минутах При каждом обычном запуске СЧПУ со автоматически сбрасывается на "0".	
Спец. для канала		

18.8 Время выполнения программы / счетчик деталей

Системная переменная	Объяснение	Активность
\$AC_OPERATING_TIME	Общее время выполнения программ ЧПУ в режиме работы АВТО в секундах Значение при каждом запуске СЧПУ со автоматически сбрасывается на "0".	- Активация через MD27860 - Только режим работы АВТО
\$AC_CYCLE_TIME	Время выполнения выбранной программы ЧПУ в секундах Значение при старте новой программы ЧПУ автоматически сбрасывается на "0".	
\$AC_CUTTING_TIME	Время обработки в секундах Измеряется время работы траекторных осей (минимум одна активна) без активного ускоренного хода во всех программах ЧПУ между NC-Start и концом программы / NC-Reset. Измерение дополнительно прерывается при активном времени ожидания. Значение при каждом запуске СЧПУ со стандартными значениями автоматически сбрасывается на "0".	
\$AC_ACT_PROG_NET_TIME	Актуальное время выполнения нетто текущей программы ЧПУ в секундах При старте программы ЧПУ автоматически сбрасывается на "0".	- Всегда активен - Только режим работы АВТО
\$AC_OLD_PROG_NET_TIME	Время выполнения нетто только что правильно завершенной с M30 программы в секундах	
\$AC_OLD_PROG_NET_TIME_COUNT	Изменения на \$AC_OLD_PROG_NET_TIME После POWER ON \$AC_OLD_PROG_NET_TIME_COUNT стоят на "0". \$AC_OLD_PROG_NET_TIME_COUNT всегда увеличивается, когда СЧПУ заново записывает \$AC_OLD_PROG_NET_TIME.	

Системная переменная	Объяснение		Активность
\$AC_PROG_NET_TIME_TRIGGER	Триггер для измерения времени выполнения:		только режим работы АВТО
	0	Нейтральное состояние Триггер не активен.	
	1	Завершение Завершает измерение и копирует значение из \$AC_ACT_PROG_NET_TIME в \$AC_OLD_PROG_NET_TIME. \$AC_ACT_PROG_NET_TIME устанавливается на "0" и затем продолжает выполняться.	
	2	Старт Запускает измерение и при этом устанавливает \$AC_ACT_PROG_NET_TIME на "0". \$AC_OLD_PROG_NET_TIME не изменяется.	
	3	Стоп Останавливает измерение. Не изменяет \$AC_OLD_PROG_NET_TIME и поддерживает \$AC_ACT_PROG_NET_TIME постоянной до продолжения.	
	4	Продолжение Продолжение измерения, т. е. остановленное ранее измерение снова возобновляется. \$AC_ACT_PROG_NET_TIME продолжает выполняться. \$AC_OLD_PROG_NET_TIME не изменяется.	
Через POWER ON все системные переменные сбрасываются на "0"!			

Примечание**Изготовитель станка**

Включение активируемых таймеров осуществляется через машинные данные MD27860 \$MC_PROCESSTIMER_MODE.

Поведение активных измерений времени при определенных функциях (например, GOTOS, процентовка = 0%, активна подача пробного хода, программный тест, ASUP, PROG_EVENT, ...) конфигурируется через машинные данные MD27850 \$MC_PROG_NET_TIMER_MODE и MD27860 \$MC_PROCESSTIMER_MODE.

Литература:

Описание функций "Основные функции"; ГПП, канал, программный режим, параметры Reset (K1), глава: Время выполнения программы

Примечание**Оставшееся время для детали**

Если последовательно изготавливаются идентичные детали, по следующим значениям времени можно определить оставшееся время для детали.

- Время обработки для последней изготовленной детали (см. \$AC_OLD_PROG_NET_TIME)
- Актуальное время обработки (см. \$AC_ACT_PROG_NET_TIME)

Оставшееся время дополнительно к текущему времени обработки отображается на интерфейсе.

Примечание**Использование STOPRE**

Системные переменные \$AC_OLD_PROG_NET_TIME и \$AC_OLD_PROG_NET_TIME_COUNT не создают не явной остановки предварительной обработки. При использовании в программе обработки детали это не критично, если значение системных переменных происходит из предшествующего хода программы. Если же триггер для измерения времени выполнения (\$AC_PROG_NET_TIME_TRIGGER) записывается высокочастотно и из-за этого \$AC_OLD_PROG_NET_TIME изменяется очень часто, то надо использовать в программе обработки детали явную STOPRE.

Граничные условия

- **Поиск кадра**
При поиске кадра время выполнения программы не определяется.
- **REPOS**
Продолжительность процесса REPOS зачисляется в текущее время обработки (\$AC_ACT_PROG_NET_TIME).

Примеры**Пример 1: Измерение длительности "mySubProgrammA"****Программный код**

```
...
N50 DO $AC_PROG_NET_TIME_TRIGGER=2
N60 FOR ii= 0 TO 300
N70 mySubProgrammA
N80 DO $AC_PROG_NET_TIME_TRIGGER=1
N95 ENDFOR
N97 mySubProgrammB
N98 M30
```

После обработки программой строки N80, в \$AC_OLD_PROG_NET_TIME стоит время выполнения нетто "mySubProgrammA".

Значение \$AC_OLD_PROG_NET_TIME:

- сохраняется после M30.
- обновляется после каждого выполнения в цикле.

Пример 2: Измерение длительности "mySubProgrammA" и "mySubProgrammC"

Программный код

```
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammD
N90 M30
```

18.8.2 Счетчики деталей

Функция "Счетчики деталей" предлагает различные счетчики, которые прежде всего могут использоваться для подсчета деталей в СЧПУ.

Счетчики существуют как специфические для канала системные переменные с доступом по записи и чтению в диапазоне значений от 0 до 999 999 999.

Системная переменная	Объяснение
\$AC_REQUIRED_PARTS	Число изготавливаемых деталей (заданное число деталей) В этом счетчике может быть определено количество деталей, при достижении которого фактическое число деталей (\$AC_ACTUAL_PARTS) сбрасывается на "0".
\$AC_TOTAL_PARTS	Общее число изготовленных деталей (общее фактическое количество) Этот счетчик указывает число всех изготовленных с момента запуска деталей. Только при запуске СЧПУ со стандартными значениями, значение автоматически сбрасывается на "0".
\$AC_ACTUAL_PARTS	Число изготовленных деталей (фактическое число деталей) В этом счетчике регистрируется число всех изготовленных с момента запуска деталей. При достижении заданного числа деталей (\$AC_REQUIRED_PARTS) счетчик автоматически сбрасывается на "0" (при условии \$AC_REQUIRED_PARTS > 0).
\$AC_SPECIAL_PARTS	Число подсчитанных пользователем деталей Этот счетчик позволяет пользователю подсчитывать детали по собственному определению. Может быть определен вывод ошибки при достижении заданного числа деталей (\$AC_REQUIRED_PARTS). Обнуление счетчика должно выполняться самим пользователем.

Примечание

Все счетчики деталей при запуске СЧПУ со стандартными значениями устанавливаются на "0" и могут считываться/записываться независимо от их активации.

Примечание

Через спец. для канала машинные данные можно управлять активацией счетчиков, моментом обнуления и алгоритмом подсчета.

Примечание**Подсчет деталей с определенной пользователем командой M**

Через машинные данные можно установить, что счетные импульсы для различных счетчиков деталей вместо запуска через конец программы M2/M30, могут быть запущены через определенные пользователем команды M.

18.9 Process DataShare - Вывод на внешнее устройство/файл (EXTOPEN, WRITE, EXTCLOSE)

Запись данных из программы обработки детали на внешнее устройство / файл осуществляется в три этапа:

1. Открыть внешнее устройство/файл
С помощью команды EXTOPEN внешнее устройство/файл открывается для канала для записи.
2. Запись данных
Выводимые данные могут быть подготовлены с помощью строковых функций языка ЧПУ, например, SPRINT. Сама запись выполняется через команду WRITE.
3. Закрыть внешнее устройство/файл
С помощью команды EXTCLOSE или при достижении конца программы (M30), а также при сбросе канала, занятые в канале внешнее устройство/файл снова разрешаются.

Синтаксис

```
DEF INT <Result>
DEF STRING[<n>] <Output>
...
EXTOPEN(<Result>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
...
<Output>=«ВЫВОД ДАННЫХ»
WRITE(<Result>,<ExtDev>,<Output>)
...
EXTCLOSE(<Result>,<ExtDev>)
```

Значение

EXTOPEN:	Предварительно определенная процедура для открытия внешнего устройства/файла	
<Result>:	Параметр 1: Переменная результата	
	На основе значения переменной результата в программе можно обработать удачный исход операции и выбрать соответствующее продолжение.	
	Тип:	INT
	Значения:	0 нет ошибок
		1 внешнее устройство не может быть открыто
		2 внешнее устройство не сконфигурировано
		3 внешнее устройство сконфигурировано с недействительным путем
		4 нет прав доступа для внешнего устройства
		5 Режим использования: внешнее устройство уже занято "эксклюзивно"
		6 Режим использования: внешнее устройство уже занято "совместно"
		7 длина файла больше LOCAL_DRIVE_MAX_FILESIZE
		8 макс. число внешних устройств превышено
		9 опция для LOCAL_DRIVE не установлена
		11 V.24-интерфейс уже занят функцией Easy-Message (только 828D)
		12 режим записи: данные противоречат extdev.ini
		16 запрограммирован недействительный внешний путь
		22 внешнее устройство не смонтировано

<ExtDev>:	Параметр 2: Символический идентификатор для открываемого внешнего устройства/файла	
	Тип:	STRING
	Символический идентификатор состоит из:	
	1. логического имени устройства	
	2. возможно с последующим путем к файлу (добавляется с "/").	
	Определены следующие логические имена устройств :	
	"LOCAL_DRIVE":	локальная карта CompactFlash (предопределено)
	"CYC_DRIVE":	указание зарезервированного диска для использования в циклах SIEMENS (предопределено)
	"/dev/ext/1",... "/dev/ext/9":	доступные сетевые диски Примечание: Требуется конфигурация в файле extdev.ini!
	"/dev/cyc/1", "/dev/cyc/2":	указание зарезервированного диска для использования в циклах SIEMENS Примечание: Требуется конфигурация в файле extdev.ini!
	"/dev/v24":	V.24-интерфейс Примечание: Требуется конфигурация в файле extdev.ini!
Путь доступа к файлу:		
- Для "LOCAL_DRIVE" и "CYC_DRIVE" должен быть указан путь доступа к файлу, к примеру: "LOCAL_DRIVE/my_dir/my_file.txt" - Логические имена устройств "/dev/ext/1...9" и "/dev/cyc/1...2" могут через конфигурацию: - уже ссылаться на файл, тогда может быть указано только логическое имя устройства, к примеру: "/dev/ext/4" - или на директорию, тогда должен быть указан путь к файлу, к примеру: "/dev/ext/5/my_dir/my_file.txt" - К "/dev/v24" путь к файлу не может быть прикреплен.		
Примечание: Прописное/строчное написание для логических имен устройств «/dev/ext/1...9», «/dev/v24» и «/dev/cyc/1...2» игнорируется, при указании пути к файлу прописное/строчное написание значимо. Для "LOCAL_DRIVE" и "CYC_DRIVE" разрешается только прописное написание.		

18.9 Process DataShare - Вывод на внешнее устройство/файл (EXTOPEN, WRITE, EXTCLOSE)

<SyncMode>:	Параметр 3: Режим обработки для команд WRITE для этого устройства/файла		
	Тип:	STRING	
	Значения:	"SYN":	Синхронная запись Выполнение программы останавливается до завершения процесса записи. Успешное завершение синхронной записи может быть проверено через обработку переменных ошибок команды WRITE.
		"ASYN":	Асинхронная запись Выполнение программы не прерывается командой WRITE. Примечание: Переменная результата команды WRITE в этом режиме бессодержательна и всегда имеет значение 0 (ошибки отсутствуют). В этом режиме отсутствует уверенность, что команда WRITE была успешной.
<AccessMode>:	Параметр 4: Режим использования для этого устройства/файла		
	Тип:	STRING	
	Значения:	"SHARED":	Устройство/файл запрошен в "совместном" режиме. Другие каналы также могут использовать этот устройство, т. е. также открыть в этом режиме.
		"EXCL":	Устройство/файл используется эксклюзивно в канале, никакой другой канал не может использовать устройства.
<WriteMode>:	Параметр 5: Режим записи для команд WRITE для этого устройства/файла (опция)		
	Тип:	STRING	
	Значения:	"APP":	Присоединение Файл сохраняется в своем содержании, вызовы записи присоединяются в конце.
		"OVR":	Перезапись Содержание файла удаляется и заново создается через последующие вызовы записи.
	Примечание: С этим параметром сконфигурированный в файле extdev.ini режим записи не может быть переписан. При конфликте вызовов EXTOPEN квитируется с ошибкой.		
WRITE:		Предварительно определенная процедура для записи выводимых данных	

EXTCLOSE:	Предварительно определенная процедура для закрытия открытого внешнего устройства/файла		
<Result>:	Параметр 1: Переменная результата		
	Тип:	INT	
	Значения:	0	нет ошибок
		16	запрограммирован недействительный внешний путь
		21	ошибка при закрытии внешнего устройства
<ExtDev>:	Параметр 2: Символический идентификатор для закрываемого внешнего устройства/файла. Описание см. EXTOPEN! Примечание: Идентификатор должен быть идентичен указанному в команде EXTOPEN идентификатору!		

Пример

Программный код

```

N10      DEF INT RESULT
N20      DEF BOOL EXTDEVICE
N30      DEF STRING[80] OUTPUT
N40      DEF INT PHASE
N50      EXTOPEN(RESULT,"LOCAL_DRIVE/my_file.txt","SYN","SHARED")
N60      IF RESULT > 0
N70          MSG("Ошибка в EXTOPEN:" << RESULT)
N80      ELSE
N90          EXTDEVICE=TRUE
N100     ENDIF
...
N200     PHASE=4
N210     IF EXTDEVICE
N220         OUTPUT=SPRINT("конец фазы: %D", PHASE)
N230         WRITE(RESULT,"LOCAL_DRIVE/my_file.txt",OUTPUT)
N240     ENDIF
...

```

См. также

Операции со строкой (Страница 91)

Записать файл (WRITE) (Страница 154)

18.10 Ошибки (SETAL)

В программе ЧПУ могут устанавливаться аварийные сообщения. Они отображаются на интерфейсе пользователя в особом поле. С аварийным сообщением связана реакция СЧПУ согласно категории аварийного сообщения.

Литература:

Дополнительную информацию по реакциям на аварийные сообщения см. Руководство по вводу в эксплуатацию.

Синтаксис

SETAL (<номер аварийного сообщения>[,<строка символов>])

Объяснение

SETAL:	Кодовое слово для программирования аварийного сообщения. SETAL должна быть запрограммирована в собственном кадре ЧПУ.	
<номер аварийного сообщения>:	Переменная типа INT. Содержит номер аварийного сообщения. Действительная область для номеров аварийных сообщений лежит между 60000 и 69999, из которых 60000 до 64999 зарезервированы для циклов SIEMENS, а 65000 до 69999 доступны пользователю.	
<строка символов>:	При программировании аварийного сообщения циклов пользователя дополнительно может быть указана строка символов с макс. 4 параметрами. В этих параметрах могут быть определены переменные тексты пользователя. Но доступны и следующие предопределенные параметры:	
	Параметр	Объяснение
	%1	Номер канала
	%2	Номер кадра, метка
	%3	Индекс текста для аварийных сообщений циклов
	%4	Дополнительный параметр аварийного сообщения

Примечание

Тексты аварийных сообщений должны быть сконфигурированы на интерфейсе пользователя.

Примечание

Если аварийное сообщение должно быть выведено на активном на интерфейсе пользователя языке, то пользователю нужна информация о текущем установленном на HMI языке. Эта информация может быть запрошена в программе обработки детали и в синхронных действиях через системную переменную \$AN_LANGUAGE_ON_HMI (см. "Текущий язык в HMI (Страница 972)").

Пример

Программный код	Комментарий
...	
N100 SETAL(65000)	; установить аварийное сообщение № 65000
...	

18.11 Расширенный останов и отвод (ESR)

Функция "Расширенный останов и отвод", в дальнейшем просто ESR, предлагает возможность гибкого реагирования на ошибки в зависимости от процесса:

- **Расширенный останов**
Если это позволяет специфическая неполадка, все разрешенные для расширенного останова оси останавливаются упорядоченно.
- **Отвод**
Находящийся в контакте инструмент отводится от детали с макс. возможной скоростью.
- **Генераторный режим (функция привода SINAMICS "Регулирование Vdc")**
При падении напряжения промежуточного контура ниже параметризуемого значения, к примеру, при отказе напряжения сети, необходимая для отвода электроэнергии вырабатывается за счет рекуперации тормозной энергии предусмотренного для этого привода (генераторный режим).

Источники запуска

Общие источники (внешн. ЧПУ/глоб. или спец. для ГПП/канала)

- Цифровые входы (к примеру, на модуле ЧПУ) или внутренний, обратно считываемый образ цифровых выходов СЧПУ (\$A_IN, \$A_OUT)
- Состояние канала (\$AC_STAT)
- VDI-сигналы (\$A_DBB)
- Сводные сообщения определенного числа аварийных сообщений (\$AC_ALARM_STAT)

Осевые источники

- Порог аварийного отвода ведомой оси (синхронный ход электронного соединения, \$VC_EG_SYNCDIFF[<ведомая ось>])
- Привод: порог предупреждения промежуточного контура (угроза падения напряжения), \$AA_ESR_STAT[<ось>]
- Привод: порог мин. скорости генератора (более нет рекуперированной энергии вращения), \$AA_ESR_STAT[<ось>].

Комбинационная логика статических синхронных действий: связь источников/реакций

Гибкая сочетаемость статических синхронных действий используется для запуска определенных относительно близко по времени реакций на основе источников.

Связь всех релевантных источников с помощью статических синхронных действий находится в руках пользователя. Он может обработать системные переменные-источники как единое целое или селективно с помощью битовых масок и привязать к ним свои желаемые реакции. Статические синхронные действия возможны во всех режимах работы.

Литература::

Описание функций "Синхронные действия"

Активация

Разрешение функции

Функции - генераторный режим, останов, отвод - разрешаются через установку соответствующего сигнала управления \$AA_ESR_ENABLE. Этот управляющий сигнал может изменяться из синхронных действий.

Запуск функции

ESR запускается совместно для всех разрешенных осей через установку системной переменной \$AC_ESR_TRIGGER.

Генераторный режим активируется в приводе "автоматически" при распознавании угрожающего падения напряжения промежуточного контура.

Автономный для привода останов и/или отвод активируются при распознавании отказа коммуникации (между ЧПУ и приводом), а также при распознавании пониженного напряжения промежуточного контура в приводе (при условии наличия конфигурирования и разрешения).

Автономный для привода останов и/или отвод также могут быть запущены со стороны ЧПУ через установку соответствующего сигнала управления \$AN_ESR_TRIGGER (команда ретрансляции на все приводы).

Литература

Подробную информацию по ESR см.:

Описание функций "Специальные функции"; Расширенный останов и отвод (R3)

18.11.1 Управляемый ЧПУ ESR

18.11.1.1 Управляемый ЧПУ отвод (POLF, POLFA, POLFMASK, POLFMLIN)

Для управляемого ЧПУ отвода должны быть выполнены определенные исходные условия (см. "Управляемый ЧПУ отвод (POLF, POLFA, POLFMASK, POLFMLIN) (Страница 708)"). Если эти условия выполнены, то для сконфигурированной(ых) в канале оси(ей) отвода через установку системной переменной \$AC_ESR_TRIGGER (или \$AA_ESR_TRIGGER для отдельных осей) активируется быстрый отвод (LIFTFAST).

Синтаксис

```
POLF [<ось>]=<позиция>  
POLFA (<ось>,<тип>,<позиция>)  
POLFMASK (<ось_1>,<ось_2>,...)  
POLFMLIN (<ось_1>,<ось_2>,...)
```

Для POLFA разрешены следующие краткие формы:

POLFA (<ось>,<тип>) ; краткая форма для отвода отдельной оси
POLFA (ось,0/1/2) ; быстрая деактивация или активация

POLFA (ось, 0, \$AA_POLFA[ось]); вызывает остановку предварительной обработки
 POLFA (ось,0); не вызывает остановки предварительной обработки

Объяснение

POLF:	адрес для указания заданного конечного положения оси отвода POLF действует модально.				
	<ось>:	имя отводимой гео-оси или оси канала/станка			
	<позиция>:	Позиция отвода			
		Тип:	REAL		
		Для гео-оси действует WCS, в иных случаях MCS. В случае идентичных идентификаторов для гео-оси или оси канала/станка отвод выполняется в WCS.			
POLFA:	предопределенный вызов подпрограммы для указания позиции отвода отдель- ных осей				
	<ось>:	Идентификатор оси канала			
	<тип>:	режим указания позиции			
		Тип:	INT		
		Значе- ние:	0:	пометить значение позиции как недействи- тельное	
			1:	значение позиции является абсолютным	
			2:	значение позиции является инкременталь- ным (дистанция)	
		Примечание: Если ось не является отдельной осью или если тип отсут- ствует/тип=0, то выводится соответствующее аварийное сообщение.			
<позиция>:	позиция отвода (см. выше)				
	Примечание: Значение позиции принимается и с типом=0. Только после это значение помечено как недействительное и должно быть запрограммировано для отвода заново.				
POLFMASK:	Предопределенный вызов подпрограммы для выбора осей, которые после за- пуска быстрого отвода должны отводиться независимо друг от друга .				
	<ось_1>,...:	Имена осей, которые при быстром отводе должны дви- гаться на свои определенные с POLF позиции. Все указанные оси должны находиться в одной системе координат.			
	POLFMASK () без указания оси деактивирует быстрый отвод для всех осей, ко- торые были отведены независимо друг от друга.				
POLFMLIN:	Предопределенный вызов подпрограммы для выбора осей, которые после за- пуска быстрого отвода должны отводиться в линейной связи .				
	<ось_1>,...:	см. выше			
	POLFMLIN () без указания оси деактивирует быстрый отвод для всех осей, ко- торые были отведены в линейной связи.				

Примечание

Перед разрешением быстрого отвода на фиксированную позицию через `POLFMASK` или `POLFMLIN`, для выбранных осей должна быть запрограммирована позиция с `POLF`.

Примечание

Если оси последовательно разрешаются с `POLFMASK`, `POLFMLIN` или `POLFMLIN`, `POLFMASK`, то для соответствующей оси всегда действует последняя установка.

Примечание

Запрограммированные с `POLF` позиции и активация через `POLFMASK` или `POLFMLIN` удаляются при старте программы обработки детали. Это означает, что пользователь в каждой программе обработки детали должен заново запрограммировать значения для `POLF` и выбранных осей в `POLFMASK` или `POLFMLIN`.

Примечание

При использовании сокращенного написания `POLFA` и изменении только типа необходимо обеспечить присваивание позиции отвода или пути отвода имеющего смысл значения! В частности, заново установить позицию отвода и путь отвода после **POWER ON**.

Пример

Отвод отдельной оси:

Программный код	Комментарий
<code>MD37500 \$MA_ESR_REACTION[AX1]=21</code>	; Управляемый ЧПУ отвод.
...	
<code>\$AA_ESR_ENABLE[AX1]=1</code>	
<code>POLFA(AX1,1,20.0)</code>	; AX1 присваивается осевая позиция отвода 20.0 (абсолютно).
<code>\$AA_ESR_TRIGGER[AX1]=1</code>	; С этого места начинается отвод.

Дополнительная информация**Условия для управляемого ЧПУ отвода**

- В канале сконфигурирована ось отвода для управляемого ЧПУ отвода:
`MD37500 $MA_ESR_REACTION = 21`
- ESR должен быть разрешен для этой оси:
`$AA_ESR_ENABLE = 1`
- Время задержки определено:
`MD21380 $MC_ESR_DELAY_TIME1`
`MD21381 $MC_ESR_DELAY_TIME2`

- В программе обработки детали спец. для оси позиции отвода запрограммированы с POLF.
- Оси с POLFMASK/POLFMLIN выбраны для управляемого ЧПУ отвода.
- Для движения отвода должны быть установлены и оставаться установленными сигналы разрешения.

Разрешение и запуск управляемого ЧПУ отвода

Если устанавливается системная переменная \$AC_ESR_TRIGGER = 1, и если в этом канале сконфигурирована ось отвода (т.е. MD37500 \$MA_ESR_REACTION = 21) и для этой оси установлено \$AA_ESR_ENABLE = 1, то в этом канале активируется быстрый отвод (LIFTFAST).

Сконфигурированное с POLF (или LFPOS) движение отвода выбранных с POLFMASK или POLFMLIN осей заменяет для этих осей определенное в программе обработки детали движение по траектории.

Для отвода предлагается макс. сумма времен MD21380 \$MC_ESR_DELAY_TIME1 и MD21381 \$MC_ESR_DELAY_TIME2. По истечении этого интервала времени и для оси отвода запускается быстрое торможение с последующим слежением.

Примечание

Расширенное движение отвода (т.е. запущенный через \$AC_ESR_TRIGGER LIFTFAST/LFPOS) **не может быть прервано** и может быть преждевременно завершено только через аварийный останов.

Примечание

Запущенный через \$AC_ESR_TRIGGER отвод заблокирован от многократного отвода.

Отвод отдельной оси

При отводе отдельной оси с POLFA должна быть запрограммирована позиция отвода отдельной оси и следующие условия должны быть соблюдены:

- \$AA_ESR_ENABLE = 1
- <ось> на момент запуска (\$AA_ESR_TRIGGER = 1) должна быть отдельной осью.
- <тип> должен быть 1 или 2.

Направление отвода при быстром отводе

На момент активации быстрого отвода учитывается действительный фрейм.

Примечание

Фреймы с вращением через POLF влияют и на направление отвода.

Переход оси

Оси отвода всегда согласованы с одним каналом ЧПУ и их переход между каналами невозможен. При попытке передачи оси отвода в другой канал сигнализируется аварийное сообщение. Только после деактивации этой оси с \$AA_ESR_ENABLE[AX] = 0

она может быть передана в новый канал. После успешного перехода оси снова могут быть нагружены с `$AA_ESR_ENABLE[AX] = 1`.

Нейтральные оси

Нейтральные оси не могут выполнять управляемый ЧПУ ESR.

18.11.1.2 Управляемый ЧПУ останов

Для сконфигурированной(ых) в канале оси(ей) останова через установку системной переменной `$AC_ESR_TRIGGER` (или `$AA_ESR_TRIGGER` для отдельных осей) можно активировать управляемый ЧПУ останов.

Условия

- В канале сконфигурирована ось останова для управляемого ЧПУ останова:
`MD37500 $MA_ESR_REACTION = 22`
- ESR должен быть разрешен для этой оси:
`$AA_ESR_ENABLE = 1`
- Время задержки определено:
`MD21380 $MC_ESR_DELAY_TIME1` (время задержки ESR-осей)
`MD21381 $MC_ESR_DELAY_TIME2` (ESR-время для интерполяционного торможения)

Процесс

В течение интервала времени в MD 21380 ось продолжает нормальную интерполяцию согласно программированию. По истечении интервала времени в MD 21380 запускается управляемое интерполяционно торможение (останов по рампе). Для управляемого интерполяционно торможения доступен макс. интервал времени в MD21381. По истечении этого интервала времени запускается быстрое торможение с последующим слежением.

Пример

Останов отдельной оси:

Программный код	Комментарий
<code>MD37500 \$MC_ESR_REACTION[AX1]=22</code>	; Управляемый ЧПУ останов.
<code>MD21380 \$MC_ESR_DELAY_TIME1[AX1]=0.3</code>	
<code>MD21381 \$MC_ESR_DELAY_TIME2[AX1]=0.06</code>	
...	
<code>\$AA_ESR_ENABLE[AX1]=1</code>	
<code>\$AA_ESR_TRIGGER[AX1]=1</code>	; С этого места начинается останов.

18.11.2 Автономный для привода ESR

18.11.2.1 Конфигурирование автономного останова привода (ESRS)

С помощью функции `ESRS ( . . . )` конфигурируются параметры привода для "остановки" автономной для привода функции ESR.

Синтаксис

`ESRS (<ось_1>, <время останова_1>[, ..., <ось_n>, <время останова_n>])`

Объяснение

<code>ESRS (. . .) :</code>	Функция для записи параметров привода для функции ESR "Останов". Функция: • должна стоять одна в кадре • вызывает остановку предварительной обработки. • не может использоваться в синхронных действиях.	
<code><ось_1>, ..., <ось_n>:</code>	Ось, для которой должен быть сконфигурирован автономный для привода останов. В приводе для этой оси записывается параметр привода p0888 (конфигурация): <code>p0888 = 1</code>	
<code><время останова_1>, ..., <время останова_n>:</code>	Тип:	AXIS
	Диапазон значений:	Идентификатор оси канала
	Период времени, в течение которого привод после возникновения ошибки продолжает постоянное движение с текущим заданным значением скорости. В приводе для этой оси записывается параметр привода p0892 (ступенчатая выдержка времени): <code>p0892 = <время останова></code>	
	Единица:	с
	Тип:	REAL
	Диапазон значений:	0.00 - 20.00
В одном вызове функции может быть запрограммировано макс. 5 осей; n = 5		

18.11.2.2 Конфигурирование автономного отвода привода (ESRS)

С помощью функции `ESRR ( . . . )` конфигурируются параметры привода для "отвода" автономной функции ESR привода.

Синтаксис

ESRR (<ось_1>, <путь отвода_1>, <скорость отвода_1> [... , <ось_n>, <путь отвода_n>, <скорость отвода_n>])

Значение

ESRR (. . .) :	Функция для записи параметров привода для функции ESR "Отвод". Функция: - должна стоять одна в кадре - вызывает остановку предварительной обработки. - не может использоваться в синхронных действиях.	
<ось_1>, ..., <ось_n>:	Ось, для которой должен быть сконфигурирован автономный для привода отвод. В приводе для этой оси записывается параметр привода p0888 (конфигурация): p0888 = 2	
	Тип:	AXIS
	Диапазон значений:	Идентификатор оси канала
<путь отвода_1>, ..., <путь отвода_n>:	Путь отвода для привода пересчитывается в скорость отвода. Значение записывается для указанной оси в параметр привода p0893 (скорость): $p0893 = (\text{путь отвода}_n \text{ пересчитанный в скорость отвода})$	
	Единица:	мм/мин, дюймов/мин, градусов/мин (в зависимости от единицы оси)
	Тип:	REAL
	Диапазон значений:	MIN - MAX
<скорость отвода_1>, ..., <скорость отвода_n>:	Скорость отвода для привода пересчитывается в промежуток времени. Значение записывается для указанной оси в параметр привода p0892 (ступенчатая выдержка времени) [сек]: $p0892 = \text{путь отвода}_n / \text{скорость отвода}_n$	
	Единица:	мм/мин, дюймов/мин, градусов/мин (в зависимости от единицы оси)
	Тип:	REAL
	Диапазон значений:	0.00 - MAX
В одном вызове функции может быть запрограммировано макс. 5 осей; n = 5		

18.12 Определение заготовки (WORKPIECE)

Система управления должна знать форму и размер заготовки, чтобы отобразить ее при графическом моделировании. Поэтому пользователь может определить заготовки через интерфейс пользователя или прямо в программе ЧПУ. Определения заготовок сохраняются и при сбросах (конца программы / канала / ГПП). При следующем запуске системы управления они автоматически удаляются.

Синтаксис

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

Объяснение

WORKPIECE(...):		Предопределенная процедура для определения заготовки	
		Остановка предварительной обработки:	да
		Автономно в кадре:	да
Параметры:			
1	"<WP>":	Название заготовки (опционально)	
		Тип данных:	STRING
		Указание необходимо, только если в одном канале могут быть несколько заготовок. При отсутствии указания автоматически принимается "WORKP<n>", при этом <n> — номер объявляющего канала.	
2	"<RefP>":	Зажим (опционально; только для фрезерных станков)	
		Тип данных:	STRING
		Диапазон значений:	"Стол" Зажим на стационарном столе
			"А" Зажим на круговой оси А
			"В" Зажим на круговой оси В
			"С" Зажим на круговой оси С
		Условие: Стол или круговая ось должны быть разрешены для зажима заготовки в соответствующих машинных данных (см. Руководство по вводу в эксплуатацию SINUMERIK Operate).	
3	"<ZeroOffset>":	Настраиваемое смещение нулевой точки для размещения заготовки (не программируется) Выбор настраиваемого смещения нулевой точки для размещения заготовки предлагается только при вводе заготовки через интерфейс пользователя. И напротив, при прямом определении заготовки в программе обработки детали заготовка всегда относится к действительному на соответствующий момент смещению нулевой точки.	

18.12 Определение заготовки (WORKPIECE)

4	"<Type>":	Форма заготовки		
		Тип данных:		STRING
		Диапазон значений:	"CYLINDER":	Цилиндр
			"PIPE":	Труба
			"RECTANGLE":	Прямоугольный параллелепипед по центру
			"BOX":	Прямоугольный параллелепипед
	"N_CORNER":	Многоугольник		
5 ... 12	<Par5> ... <Par12>:	Параметры описания формы заготовки		
		Тип данных:		REAL
		Количество необходимых параметров и их значение зависят от формы соответствующей заготовки и значения параметра битового формата.		
		См.: • Таблица "Параметры описания формы заготовки" • Таблица "Параметры битового формата"		
WORKPIECE () :		Вызов WORKPIECE без параметров удаляет все определения заготовок.		
WORKPIECE (<WP>) :		Вызов WORKPIECE с именами деталей удаляет только эти определения заготовок.		

Таблица 18-1 Параметры описания формы заготовки

Форма заготовки	Параметр							
	<Par5>	<Par6>	<Par7>	<Par8>	<Par9>	<Par10>	<Par11>	<Par12>
Цилиндр	Параметры битового формата Действительное значение,	Исходная точка Z_0	Длина Z_1	Размер обработки Z_B	Наружный диаметр d_0	-	Вращение вокруг круговой оси	-
Труба	интерпретируемое как кодированное в битах целочисленное число. Биты определяют значение следующих параметров (см. таблицу "Параметры битового формата").	Исходная точка Z_0	Длина Z_1	Размер обработки Z_B	Наружный диаметр d_0	Толщина стенки (инкр.) / внутренний диаметр d_1 (абс.)	Вращение вокруг круговой оси	-
Прямоугольный параллелепипед по центру		Исходная точка Z_0	Длина Z_1	Размер обработки Z_B	Ширина W	Длина L	Вращение вокруг круговой оси	-
Прямоугольный параллелепипед		Исходная точка Z_0	Длина Z_1	Размер обработки Z_B	X_0	Y_0	X_1	Y_1
Многоугольник		Исходная точка Z_0	Длина Z_1	Размер обработки Z_B	Количество углов	Размер ключа	Вращение вокруг круговой оси	-

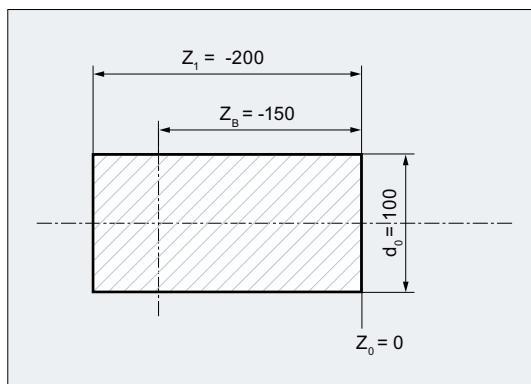
Таблица 18-2 Параметры битового формата

Бит	Объяснение	
4 (0x0010)	Прямоугольный параллелепипед: X_1	
	=0	инкр.
	=1	абс.
5 (0x0020)	Прямоугольный параллелепипед: Y_1	
	=0	инкр.
	=1	абс.
6 (0x0040)	Длина Z_1 (конечный размер)	
	=0	инкр.
	=1	абс.
Бит 7 (0x0080)	Размер обработки Z_B	
	=0	инкр.
	=1	абс.
Бит 8 (0x0100)	Труба: Толщина стенки / Внутренний диаметр	
	=0	инкр.
	=1	абс.

Бит	Объяснение	
9 (0x0200)	Многоугольник	
	=0	Размер ключа
	=1	Длина грани
12 (0x1000)	Зажим для токарных станков	
	=0	Главный шпиндель
	=1	Противошпиндель
13 (0x2000)	Противошпиндель	
	=0	с отражением
	=1	без отражения

Примеры

Пример 1: Цилиндрическая заготовка на токарном станке



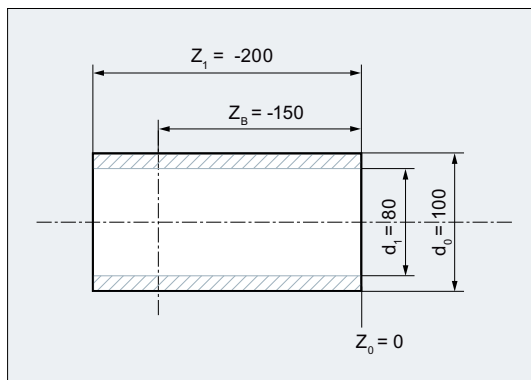
Программный код

```
...
WORKPIECE(,,, "CYLINDER", 0, 0, -200, -150, 100)
...
```

Комментарий

```
; Определение заготовки:
; Форма заготовки: Цилиндр
; Параметр битового формата =0 (бит не
установлен) → Значения длины и размера
обработки — инкрементальные, заготовка
на главном шпинделе
; Исходная точка (Z0) =0
; Длина (Z1) =-200
; Размер обработки (ZB) =-150
; Наружный диаметр (d0) =100
```

Пример 2: Трубообразная заготовка на токарном станке



Программный код

```
...
WORKPIECE( , , "PIPE", 256, 0, -200, -150, 100, 80)

```

```
...
```

Комментарий

; **Определение заготовки:**
 ; Форма заготовки: Труба
 ; Параметр битового формата =256(бит 8=1) → Значение внутреннего диаметра — абсолютное, значения длины и размера обработки — инкрементальные, заготовка на главном шпинделе
 ; Исходная точка (Z0) = 0
 ; Длина (Z1) = -200
 ; Размер обработки (ZB) = -150
 ; Наружный диаметр (d0) = 100
 ; Внутренний диаметр (d1) = 80

18.13 Переключение языкового режима (G290, G291)

Система управления позволяет считывать и обрабатывать программы обработки детали с внешних систем CNC. Для этого необходимо при вводе в эксплуатацию установить соответствующий языковой режим ЧПУ (диалект ISO).

Литература:

Описание функций "Диалекты ISO"

Режим диалекта ISO можно активировать отдельно для каждого канала. Например, канал 1 может идти в режиме диалекта ISO, канал 2 в режиме SINUMERIK.

Переключение с режима SINUMERIK на режим диалекта ISO выполняется в программе ЧПУ командами G-группы 47. На активный инструмент, коррекции на инструмент и смещения нулевой точки переключение не влияет.

Синтаксис

```
G291
...
G290
```

Объяснение

G290:	Активировать языковой режим SINUMERIK	
	Автономно в кадре:	да
	Активность:	модально
G291:	Активировать языковой режим ISO	
	Автономно в кадре:	да
	Активность:	модально

Условия

Режим SINUMERIK

- Через машинные данные можно сохранить предустановки G-команд каждого канала.
- В режиме SINUMERIK невозможно программировать языковые команды диалектов ISO.

Режим диалекта ISO

- Через машинные данные можно настроить режим диалекта ISO в качестве установки системы управления по умолчанию. В этом случае система управления запускается в режиме диалекта ISO.
- Возможно программирование только G-команд диалекта ISO. Программировать функции SINUMERIK в режиме диалекта ISO невозможно.
- Комбинировать диалекта ISO и языка SINUMERIK в одном кадре ЧПУ невозможно.
- Переключение через G-команды между диалектом M ISO (фрезерная обработка) и диалектом T ISO (токарная обработка) невозможно.

- Возможен вызов подпрограмм, запрограммированных в режиме SINUMERIK.
- При необходимости использования функций SINUMERIK необходимо сначала переключиться в режим SINUMERIK. (см. пример).

Пример

Сжатие линейных кадров в режиме диалекта ISO

Программный код	Комментарий
N5 G290	; Активировать языковой режим SINUMERIK
N10 COMPON	; COMPON — команда языка Siemens, которая активирует функцию сжатия, заменяющую последовательные линейные кадры на полиномиальные кадры с максимально длинными траекториями.
N15 G291	; Активировать языковой режим ISO.
N20 G01 X100 Y100 F1000	; Поскольку COMPON была активирована в режиме SINUMERIK, она способна сжимать и линейные кадры в режиме диалекта ISO.
...	

Собственные программы обработки резаньем

19.1 Функции поддержки для обработки резаньем

Для обработки резаньем предлагаются готовые циклы обработки. Кроме этого, с помощью перечисленных ниже функций можно создавать собственные программы обработки резаньем:

- Создание таблицы контуров (CONTPRON)
- Создание кодированной таблицы контуров (CONTDCON)
- Отключить подготовку контура (EXECUTE)
- Определить точку пересечения между двумя элементами контура (INTERSEC) (только для таблиц, созданных через CONTPRON).
- Покадровая обработка элементов контура таблицы (EXECTAB) (только для таблиц, созданных через CONTPRON).
- Вычисление данных окружности (CALCDAT)

Примечание

Эти функции могут использоваться не только для обработки резаньем, но и универсально.

Условия

Перед вызовом функций CONTPRON или CONTDCON должен быть:

- выполнен подвод к стартовой точке, обеспечивающей обработку без столкновений.
- коррекция радиуса резцов с G40 должна быть отключена.

19.2 Создание таблицы контуров (CONTPRON)

С помощью CONTPRON включается подготовка контура. Вызванные после кадры ЧПУ не выполняются, а разбиваются на отдельные движения и сохраняются в таблицу контура. Каждому элементу контура соответствует строка в таблице в двухмерном поле таблицы контуров. Возвращается число полученных поднутрений.

Синтаксис

Включить подготовку контура:

CONTPRON (<таблица контура>, <режим обработки>, <поднутрения>, <направление обработки>)

Отключить подготовку контура и одновременно вернуться в обычный режим обработки:

EXECUTE (<ОШИБКА>)

См. " Отключить подготовку контура (EXECUTE) (Страница 739) "

Объяснение

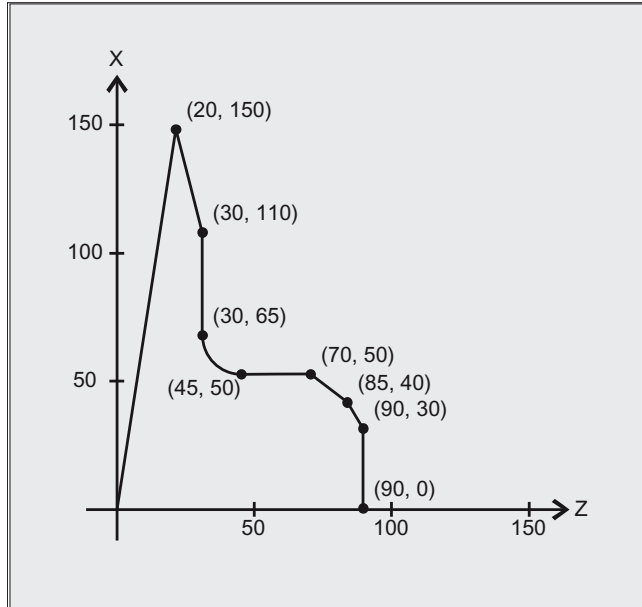
CONTPRON:	Предопределенная процедура для включения подготовки контура для создания таблицы контура		
<таблица контура>:	Имя таблицы контура		
<режим обработки>:	Параметры для режима обработки		
	Тип:	CHAR	
	Значение:	"G":	Продольная обточка: внутренняя обработка
		"L":	Продольная обточка: наружная обработка
		"N":	Торцевание: внутренняя обработка
"P":		Торцевание: наружная обработка	
<поднутрения>:	Переменная результата для числа возникших элементов поднутрения		
	Тип:	INT	
<направление обработки>:	Параметры для направления обработки		
	Тип:	INT	
	Значение:	0	Подготовка контура вперед (стандартное значение)
		1	Подготовка контура в обоих направлениях

Пример 1

Создание таблицы контура с:

- именем "КТАВ"
- макс. 30 элементами контура (окружности, прямые)

- переменной для числа встречающихся элементов поднутрений
- переменной для сообщений об ошибках



Программа ЧПУ:

Программный код	Комментарий
N10 DEF REAL KTAB[30,11]	; Таблица контура с именем KTAB и макс. 30 элементами контура, значение параметра 11 (число колонок таблицы) это фиксированная величина.
N20 DEF INT ANZHINT	; Переменная для числа элементов поднутрений с именем ANZHINT.
N30 DEF INT FEHLER	; Переменная для квитирования ошибок (0=нет ошибок, 1=ошибка).
N40 G18	
N50 CONTPRON (KTAB,"G",ANZHINT)	; Включить подготовку контура.
N60 G1 X150 Z20	; N60 до N120: Описание контура
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE (FEHLER)	; Завершить заполнение таблицы контура, переключение на обычный программный режим.
N140 ...	; Продолжить обработку таблицы.

Таблица контура КТАВ:

Ин- декс Стро- ка	Графа									
(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
7	7	11	0	0	20	150	0	82.40535663	0	0
0	2	11	20	150	30	110	-1111	104.0362435	0	0
1	3	11	30	110	30	65	0	90	0	0
2	4	13	30	65	45	50	0	180	45	65
3	5	11	45	50	70	50	0	0	0	0
4	6	11	70	50	85	40	0	146.3099325	0	0
5	7	11	85	40	90	30	0	116.5650512	0	0
6	0	11	90	30	90	0	0	90	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Объяснение содержания колонок:

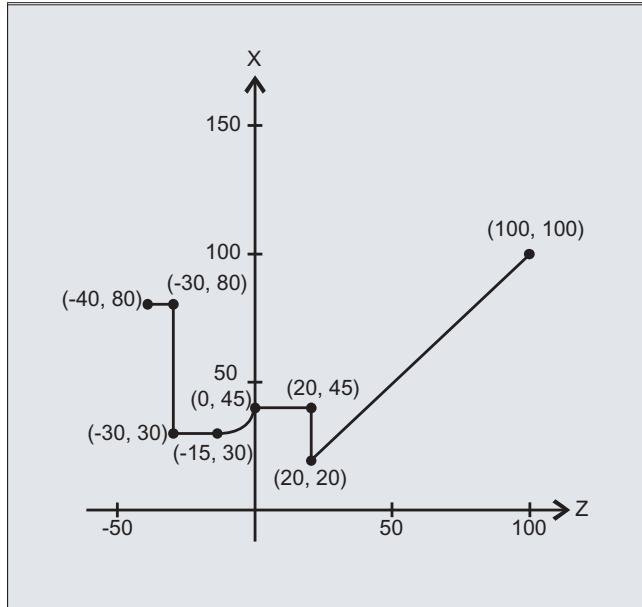
- (0) Указатель на следующий элемент контура (на номер строки таковой)
- (1) Указатель на предшествующий элемент контура
- (2) Кодировка режима контура для движения
Возможные значения для $X = abc$
 $a = 10^2$ $G90 = 0$ $G91 = 1$
 $b = 10^1$ $G70 = 0$ $G71 = 1$
 $c = 10^0$ $G0 = 0$ $G1 = 1$ $G2 = 2$ $G3 = 3$
- (3), (4) Начальная точка элементов контура
(3) = абсцисса, (4) = ордината в актуальной плоскости
- (5), (6) Конечная точка элемента контура
(5) = абсцисса, (6) = ордината в актуальной плоскости
- (7) Указатель макс./мин.: обозначает локальные максимум и минимум на контуре
- (8) Макс. значение между элементом контура и абсциссой (при продольной обработке) или ординатой (при поперечной обработке). Угол зависит от запрограммированного режима обработки.
- (9), (10) координаты центра элемента контура, если это круговой кадр.
(9) = абсцисса, (10) = ордината

Пример 2

Создание таблицы контура с

- именем КТАВ
- макс. 92 элементами контура (окружности, прямые)

- режим работы: продольная обточка, наружная обработка
- подготовка вперед и назад



Программа ЧПУ:

Программный код	Комментарий
N10 DEF REAL KTAB[92.11]	; Таблица контура с именем KTAB и макс. 92 элементами контура, значение параметра 11 это фиксированная величина.
N20 DEF CHAR BT="L"	; Режим работы для CONTPRON: продольная обточка, наружная обработка
N30 DEF INT HE=0	; Число элементов поднутрения=0
N40 DEF INT MODE=1	; Подготовка вперед и назад
N50 DEF INT ERR=0	; Квитирование ошибок
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(KTAB,BT,HE,MODE)	; Включить подготовку контура.
N110 G1 G90 Z20 X20	
N120 X45	
N130 Z0	
N140 G2 Z-15 X30 K=AC(-15) I=AC(45)	
N150 G1 Z-30	
N160 X80	
N170 Z-40	
N180 EXECUTE(ERR)	; Завершить заполнение таблицы контура, переключение на обычный программный режим.
...	

Таблица контура КТАВ:

После завершения подготовки контура он доступен в обоих направлениях.

Индекс	Колонка										
Строка	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
0	6 ¹⁾	7 ²⁾	11	100	100	20	20	0	45	0	0
1	0 ³⁾	2	11	20	20	20	45	-3	90	0	0
2	1	3	11	20	45	0	45	0	0	0	0
3	2	4	12	0	45	-15	30	5	90	-15	45
4	3	5	11	-15	30	-30	30	0	0	0	0
5	4	7	11	-30	30	-30	45	-1111	90	0	0
6	7	0 ⁴⁾	11	-30	80	-40	80	0	0	0	0
7	5	6	11	-30	45	-30	80	0	90	0	0
8	1 ⁵⁾	2 ⁶⁾	0	0	0	0	0	0	0	0	0
	...										
83	84	0 ⁷⁾	11	20	45	20	80	0	90	0	0
84	90	83	11	20	20	20	45	-1111	90	0	0
85	0 ⁸⁾	86	11	-40	80	-30	80	0	0	0	0
86	85	87	11	-30	80	-30	30	88	90	0	0
87	86	88	11	-30	30	-15	30	0	0	0	0
88	87	89	13	-15	30	0	45	-90	90	-15	45
89	88	90	11	0	45	20	45	0	0	0	0
90	89	84	11	20	45	20	20	84	90	0	0
91	83 ⁹⁾	85 ¹⁰⁾	11	20	20	100	100	0	45	0	0

Объяснение содержания колонок и примечаний к строкам 0, 1, 6, 8, 83, 85 и 91

Действуют названные в примере 1 пояснения содержания колонок.

Всегда в строке таблицы 0:

- 1) Предшествующий элемент: строка n содержит конец контура вперед
- 2) Следующий элемент: строка n это конец таблицы контура вперед

По одному разу в элементах контура вперед:

- 3) Предшествующий элемент: начало контура (вперед)
- 4) Следующий элемент: конец контура (вперед)

Всегда на строку конца таблицы контура (вперед) +1:

- 5) Предшествующий элемент: число поднутрений вперед
- 6) Следующий элемент: число поднутрений назад

По одному разу в элементах контура назад:

- 7) Следующий элемент: конец контура (назад)
- 8) Предшествующий элемент: начало контура (назад)

Всегда в последней строке таблицы:

- 9) Предшествующий элемент: строка n это начало таблицы контура (назад)

10) Следующий элемент: строка n содержит начало контура (назад)

Дополнительная информация

Разрешенные команды движения, система координат

Для программирования контура разрешены следующие команды G:

- G-группа 1: G0, G1, G2, G3

Дополнительно возможны:

- Закругление и фаска
- Программирование окружности через CIP и CT

Функции "Сплайн", "Полином" и "Резьба" приводят к ошибкам.

Изменения системы координат через включение фрейма между CONTPRON и EXECUTE не допускаются. То же самое действительно для переключения между G70 и G71 или между G700 и G710.

Переход гео-осей с GEOAX при подготовке таблицы контура приводит к ошибке.

Элементы поднутрения

Описание контура отдельных элементов поднутрения может осуществляться по выбору в подпрограмме или в отдельных кадрах.

Обработка резаньем независимо от запрограммированного направления контура

Подготовка контура с CONTPRON была расширена таким образом, что после ее вызова таблица контура доступна независимо от запрограммированного направления.

19.3 Создание кодированной таблицы контуров (CONTDCON)

При включенной с CONTDCON подготовкой контура вызванные после кадры ЧПУ сохраняются в таблице контура с 6 колонками в удобной для памяти кодировке. Каждому элементу контура соответствует строка таблицы контура. Зная указанные ниже правила кодировки можно, например, создавать для циклов из строк таблицы программы кода DIN. В строке таблицы с номером 0 сохраняются данные исходной точки.

Синтаксис

Включить подготовку контура:

CONTDCON (<таблица контура>, <направление обработки>)

Отключить подготовку контура и одновременно вернуться в обычный режим обработки:

EXECUTE (<ОШИБКА>)

См. " Отключить подготовку контура (EXECUTE) (Страница 739) "

Объяснение

CONTDCON:	Предопределенная процедура для включения подготовки контура для создания кодированной таблицы контура		
<таблица контура>:	Имя таблицы контура		
<направление обработки>:	Параметры для направления обработки		
	Тип:	INT	
	Значение:	0	Подготовка контура согласно последовательности кадров контура (стандартное значение)
		1	недопустимо

Примечание

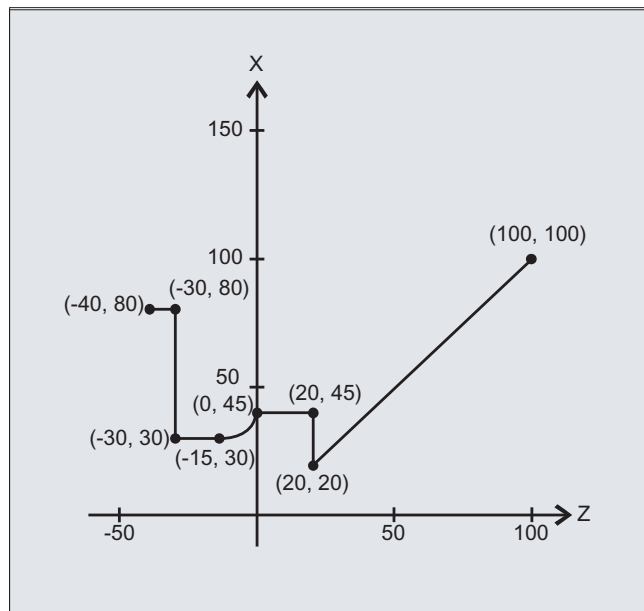
Разрешенные для CONTDCON G-команды в табулируемом сегменте программы объемнее, чем при CONTPRON. Кроме этого, также сохраняются подачи и тип подачи на участок контура.

Пример

Создание таблицы контура с:

- именем "КТАВ"
- элементами контура (окружности, прямые)
- Режим работы: токарная обработка
- Направление обработки: вперед

19.3 Создание кодированной таблицы контуров (CONTDCON)



Программа ЧПУ:

Программный код	Комментарий
N10 DEF REAL KTAB[9.6]	; Таблица контура с именем KTAB и 9 строками. Они допускают 8 кадров контура. Значение параметра 6 (число колонок в таблице) это фиксированная величина.
N20 DEF INT MODE = 0	; Переменная для направления обработки. Стандартное значение 0: только в запрограммированном направлении контура.
N30 DEF INT ERROR = 0	; Переменная для квитирования ошибок.
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	; Вызов подготовки контура (MODE можно опустить).
N110 G1 Z20 X20 F200	; Описание контура.
N120 G9 X45 F300	
N130 Z0 F400	
N140 G2 Z-15 X30 K=AC(-15) I=AC(45) F100	
N150 G64 Z-30 F600	
N160 X80 F700	
N170 Z-40 F800	
N180 EXECUTE(ERROR)	; Завершить заполнение таблицы контура, переключение на обычный программный режим.
...	

Таблица контура КТАВ:

	Индекс колонки					
	0	1	2	3	4	5
Индекс строки	Режим контура	Конечная точка абсцисса	Конечная точка ордината	Центр абсцисса	Центр ордината	Подача
0	30	100	100	0	0	7
1	11031	20	20	0	0	200
2	111031	20	45	0	0	300
3	11031	0	45	0	0	400
4	11032	-15	30	-15	45	100
5	11031	-30	30	0	0	600
6	11031	-30	80	0	0	700
7	11031	-40	80	0	0	800
8	0	0	0	0	0	0

Объяснение содержания колонок:Строка 0: кодировки для **стартовой точки**:

- Колонка 0: 10^0 (первая позиция): $G0 = 0$
 10^1 (вторая позиция): $G70 = 0$, $G71 = 1$, $G700 = 2$, $G710 = 3$
- Колонка 1: стартовая точка абсцисса
- Колонка 2: стартовая точка ордината
- Колонка 3-4: 0
- Колонка 5: индекс строки последнего участка контура в таблице

Строки 1-п: записи **участков контура**

- Колонка 0: 10^0 (первая позиция): $G0 = 0$, $G1 = 1$, $G2 = 2$, $G3 = 3$
 10^1 (вторая позиция): $G70 = 0$, $G71 = 1$, $G700 = 2$, $G710 = 3$
 10^2 (третья позиция): $G90 = 0$, $G91 = 1$
 10^3 (четвертая позиция): $G93 = 0$, $G94 = 1$, $G95 = 2$, $G96 = 3$
 10^4 (пятая позиция): $G60 = 0$, $G44 = 1$, $G641 = 2$, $G642 = 3$
 10^5 (шестая позиция): $G9 = 1$
- Колонка 1: конечная точка абсцисса
- Колонка 2: конечная точка ордината
- Колонка 3: центр абсцисса при круговой интерполяции
- Колонка 4: центр ордината при круговой интерполяции
- Колонка 5: подача

Дополнительная информация

Разрешенные команды движения, система координат

Для программирования контура разрешены следующие группы G и команды G:

G-группа 1:	G0, G1, G2, G3
G-группа 10:	G60, G64, G641, G642
G-группа 11:	G9
G-группа 13:	G70, G71, G700, G710
G-группа 14:	G90, G91
G-группа 15:	G93, G94, G95, G96, G961

Дополнительно возможны:

- Закругление и фаска
- Программирование окружности через CIP и CT

Функции "Сплайн", "Полином" и "Резьба" приводят к ошибкам.

Изменения системы координат через включение фрейма между CONTDCON и EXECUTE не допускаются. То же самое действительно для переключения между G70 и G71 или между G700 и G710.

Переход гео-осей с GEOAX при подготовке таблицы контура приводит к ошибке.

Направление обработки

Создаваемая с помощью CONTDCON таблица контура предусмотрена для обработки резанием в запрограммированном направлении контура.

19.4 Определить точку пересечения между двумя элементами контура (INTERSEC)

INTERSEC определяет точку пересечения двух нормированных элементов контура из созданных с помощью CONTPRON таблиц контуров.

Синтаксис

```
<состояние>=INTERSEC (<таблица контура_1>[<элемент контура_1>],
<таблица контура_2>[<элемент контура_2>],<точка пересечения>,<режим обработки>)
```

Объяснение

INTERSEC:	Предопределенная функция для определения точки пересечения двух элементов контура из созданных с помощью CONTPRON таблиц контуров		
<состояние>:	Переменная для состояния точки пересечения		
	Тип:	BOOL	
	Значение:	TRUE	точка пересечения найдена
		false	точка пересечения не найдена
<таблица контура_1>:	имя первой таблицы контура		
<элемент контура_1>:	номер элемента контура первой таблицы контура		
<таблица контура_2>:	имя второй таблицы контура		
<элемент контура_2>:	номер элемента контура второй таблицы контура		
<точка пересечения>:	координаты точки пересечения в активной плоскости (G17 / G18 / G19)		
	Тип:	REAL	
<режим обработки>:	параметры для режима обработки		
	Тип:	INT	
	Значение:	0	вычисление точки пересечения в активной с параметром 2 плоскости (стандартное значение)
		1	вычисление точки пересечения независимо от переданной плоскости

Примечание

Учитывать, что переменные перед их использованием должны быть определены.

Для передачи контуров необходимо соблюдать определенные с CONTPRON значения:

Параметры	Значение
2	Кодировка режима контура для движения
3	Начальная точка контура абсцисса
4	Начальная точка контура ордината
5	Конечная точка контура абсцисса
6	Конечная точка контура ордината

19.4 Определить точку пересечения между двумя элементами контура (INTERSEC)

Параметры	Значение
9	Координата центра для абсциссы (только для кругового контура)
10	Координата центра для ординаты (только для кругового контура)

Пример

Определить точку пересечения элемента контура 3 таблицы TABNAME1 и элемента контура 7 таблицы TABNAME2. Координаты точки пересечения в активной плоскости сохраняются в переменной SCHNITT (1-й элемент = абсцисса, 2-й элемент = ордината). Если точки пересечения не существует, то выполняется переход на KEINSCH (точка пересечения не найдена).

Программный код	Комментарий
DEF REAL TABNAME1[12,11]	; Таблица контура 1
DEF REAL TABNAME2[10,11]	; Таблица контура 2
DEF REAL ISCOORD[2]	; Переменная для координат точки пересечения.
DEF BOOL ISPOINT	; Переменная для состояния точки пересечения.
DEF INT MODE	; Переменная для режима обработки.
...	
MODE=1	; Вычисление независимо от активной плоскости.
N10 ISPOINT=INTERSEC(TABNAME1[3], TABNAME2[7], ISCOORD, MODE)	; Вызов точки пересечения элементов контура.
N20 IF ISPOINT==FALSE GOTO KEINSCH	; Переход на KEINSCH.
...	

19.5 Покадровая обработка элементов контура таблицы (ЕХЕСТАВ)

С помощью ЕХЕСТАВ возможна покадровая обработка элементов контура таблицы, которые, к примеру, были созданы с помощью CONTRON.

Синтаксис

ЕХЕСТАВ (<таблица контура>[<элемент контура>])

Объяснение

ЕХЕСТАВ:	Предопределенная процедура для обработки элемента контура
<таблица контура>:	Имя таблицы контура
<элемент контура>:	Номер элемента контура

Пример

Элементы контура 0 до 2 таблицы КТАВ должны быть обработаны покадрово.

Программный код	Комментарий
N10 ЕХЕСТАВ (КТАВ[0])	; Перемещение элемента 0 таблицы КТАВ.
N20 ЕХЕСТАВ (КТАВ[1])	; Перемещение элемента 1 таблицы КТАВ.
N30 ЕХЕСТАВ (КТАВ[2])	; Перемещение элемента 2 таблицы КТАВ.

19.6 Вычисление данных окружности (CALCDAT)

С помощью CALCDAT из трех или четырех известных точек окружности можно вычислить радиус и координаты центра окружности. Указанные точки не должны совпадать.

В случае 4 точек, расположенных не точно на окружности, для центра окружности и радиуса формируется среднее значение.

Примечание

Правила вычисления для формирования среднего значения

Вычисление дуги окружности выполняется 4 раза:

1. с точками окружности 1, 2, 3
2. с точками окружности 1, 2, 4
3. с точками окружности 1, 3, 4
4. с точками окружности 2, 3, 4

Координаты центра окружности (значение абсциссы и значение ординаты) вычисляются через сложение значений абсциссы/ординаты, полученных в результате четырех вычислений дуги окружности, и деление суммы на 4.

Для вычисления радиуса берется корень из суммы четырех радиусов, полученных при вычислениях дуги окружности, и результат умножается на 0,5.

Синтаксис

<состояние>=CALCDAT (<точки окружности>[<число>,<вид>],<число>,<результат>)

Объяснение

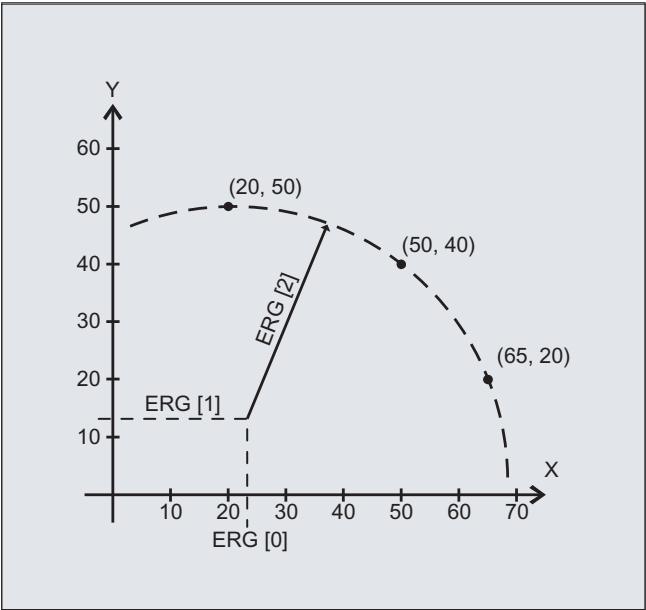
CALCDAT:	Предопределенная функция для вычисления радиуса и координат центра окружности из 3 или 4 точек		
<состояние>:	Переменная для состояния вычисления окружности		
	Тип:	BOOL	
	Значение:	TRUE	Указанный точки лежат на одной окружности.
false		Указанный точки не лежат на одной окружности.	
<точки окружности>[]:	Переменная для указания точек окружности с параметрами:		
	<число>:	Число точек окружности (3 или 4)	
	<вид>:	Вид указания координат, к примеру, 2 для 2-точечных координат	
<число>:	Параметр для числа используемых для вычисления точек (3 или 4)		
<результат>[3]:	Переменная для результата:		
	Указание координат центра окружности и радиуса		
	0	Координата центра окружности: значение абсциссы	
	1	Координата центра окружности: значение ординаты	
	2	Радиус	

Примечание

Учитывать, что переменные перед их использованием должны быть определены.

Пример

Из трех точек необходимо определить, лежат ли они на круговом сегменте.



Программный код	Комментарий
N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)	; Переменная для указания точек окружности
N20 DEF REAL ERG[3]	; Переменная для результата:
N30 DEF BOOL STATUS	; Переменная для состояния.
N40 STATUS=CALCDAT(PKT,3,ERG)	; Вызов полученных данных окружности.
N50 IF STATUS == FALSE GOTOF ERROR	; Переход на ошибку.

19.7 Отключить подготовку контура (EXECUTE)

С помощью EXECUTE подготовка контура отключается и одновременно происходит возврат в обычный режим обработки.

Синтаксис

EXECUTE (<ОШИБКА>)

Объяснение

EXECUTE:	Предопределенная процедура для завершения подготовки контура	
<ошибка>:	Переменная для квитирования ошибки	
	Тип:	INT
	Значение переменной показывает, был ли контур подготовлен без ошибок:	
	0	ошибка
	1	нет ошибок

Пример

Программный код

```
...  
N30 CONTPRON (...)  
N40 G1 X... Z...  
...  
N100 EXECUTE (...)  
...
```


Внешнее программирование циклов

20.1 Технологические циклы

20.1.1 Введение

Содержание

Эта глава содержит описание циклов для технологий токарной обработки, фрезерования, шлифования.

Структура

Описание цикла построено следующим образом:

- **Синтаксис**
Имя цикла и последовательность вызова передаваемых параметров
- **Параметры**
Таблица для объяснения отдельных параметров

Описание параметров

В таблице указаны следующие данные для параметра: Имя, описание, диапазон значений и зависимости относительно других параметров.

Графа для ссылки на параметр в маске облегчает нахождение запрограммированных на СЧПУ значений при обратном переводе вызовов циклов, созданных на внешних устройствах.

Параметр "только для интерфейса"

В таблице параметры обозначены как "только для интерфейса". Они не имеют значения для функций цикла. Они служат только для возможности полного обратного перевода вызовов циклов. Если они не запрограммированы, то обратный перевод цикла все же возможен, в этом случае поля обозначены соответствующими цветами и должны быть заполнены в маске.

Параметры "зарезервировано"

Параметры, описанные с "зарезервировано", должны быть запрограммированы со значением 0 или с незначимой запятой, чтобы согласование следующих параметров вызова с внутренними параметрами цикла совпадало. Исключение: для строковых параметров Значение "" или незначимая запятая.

Повторение циклов на образце позиции

Сверлильные и фрезерные циклы могут повторяться на образцах позиций (модальные вызовы). В этом случае записать перед циклом в той же строке `MCALL`, к примеру, `MCALL CYCLE83 ( . . . )`.

Примечание

Если определенные передаваемые параметры (к примеру, `<_VARI>`, `<_GMODE>`, `<_DMODE>`, `<_AMODE>`) запрограммированы косвенно как параметры, маска ввода при обратном переводе открывается, но не может быть сохранена, т. к. нет однозначного согласования с определенными полями выбора.

20.1.2 Обзор по технологиям

В следующей обзорной таблице перечислены все доступные технологические циклы с возможностью внешнего программирования с разбивкой по соответствующим технологиям:

Технология	Технологический цикл
Сверление	• CYCLE81 - сверление, центрование (Страница 784) • CYCLE82 - сверление, цекование (Страница 785) • CYCLE85 - развертывание (Страница 793) • CYCLE86 - растачивание (Страница 794) • CYCLE83 - глубокое сверление (Страница 787) • CYCLE830 - Глубокое сверление 2 (Страница 818) • CYCLE84 - нарезание внутренней резьбы без компенсирующего патрона (Страница 790) • CYCLE840 - Нарезание внутренней резьбы с компенсирующим патроном (Страница 829) • CYCLE78 - Сверлильное резьбофрезерование (Страница 778) • CYCLE802 - любые позиции (Страница 815) • HOLES1 - Ряд отверстий (Страница 744) • CYCLE801 - Решетка или рамка (Страница 814) • HOLES2 - Окружность отверстий (Страница 745)
Токарная обработка	• CYCLE951 - обработка резанием (Страница 840) • CYCLE930 - выточка (Страница 834) • CYCLE940 - формирование канавки (Страница 837) • CYCLE99 - нарезание резьбы резцом (Страница 803) • CYCLE98 - цепочка резьб (Страница 799) • CYCLE92 - отрез (Страница 796)

Технология	Технологический цикл
Контурная обточка	• CYCLE62 - Вызов контура (Страница 764) • CYCLE952 - прорезание контура (Страница 843)
Фрезерная обработка	• CYCLE61 - плоское фрезерование (Страница 762) • POCKET3 - Фрезерование прямоугольного кармана (Страница 747) • POCKET4 - Фрезерование кругового кармана (Страница 749) • CYCLE76 - фрезерование прямоуг. цапфы (Страница 774) • CYCLE77 - фрезерование круговой цапфы (Страница 776) • CYCLE79 - многогранник (Страница 782) • SLOT1 - Продольный паз (Страница 752) • SLOT2 - кольцевая канавка (Страница 755) • CYCLE899 - фрезерование открытого паза (Страница 831) • LONGHOLE - Продольный паз (Страница 757) • CYCLE70 - резьбофрезерование (Страница 769) • CYCLE60 - Цикл гравирования (Страница 759)
Контурное фрезерование	• CYCLE62 - Вызов контура (Страница 764) • CYCLE72 - фрезерование траектории (Страница 770) • CYCLE63 - фрезерование кармана (Страница 765) • CYCLE64 - Предварительное сверление кармана (Страница 767)
Шлифование	• CYCLE495 - Профилирование (Страница 808) • CYCLE435 - установка системы координат правящего инструмента (Страница 808) • CYCLE4071 - Продольное шлифование с подачей в точке возврата (Страница 849) • CYCLE4072 - Продольное шлифование с подачей в точке возврата и сигналом отмены (Страница 850) • CYCLE4073 - Продольное шлифование с непрерывной подачей на глубину (Страница 854) • CYCLE4074 - Продольное шлифование с непрерывной подачей на глубину и сигналом отмены (Страница 855) • CYCLE4075 - Плоское шлифование с подачей в точке возврата (Страница 858) • CYCLE4077 - Плоское шлифование с подачей в точке возврата и сигналом отмены (Страница 861) • CYCLE4078 - Плоское шлифование с непрерывной подачей на глубину (Страница 864) • CYCLE4079 - Плоское шлифование с прерывистой подачей на глубину (Страница 866)

Технология	Технологический цикл
Прочее	• CYCLE800 - поворот (Страница 810) • CYCLE832 - High Speed Settings (Страница 825)
Все	• GROUP_BEGIN — Начало блока программы (Страница 868) • GROUP_END — Конец блока программы (Страница 869) • GROUP_ADDEND — конец дополнения ввода (Страница 869)

20.1.3 HOLES1 - Ряд отверстий

Синтаксис

HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>, <_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
1	X0	<SPCA>	REAL	Исходная точка для ряда отверстий в 1-й оси (абс.)		
2	Y0	<SPCO>	REAL	Исходная точка для ряда отверстий в 2-й оси (абс.)		
3	α0	<STA1>	REAL	Базовый угол поворота (угол к 1-й оси)		
4	L0	<FDIS>	REAL	Расстояние от 1-го отверстия до исходной точки		
5	I	<DBH>	REAL	Расстояние между отверстиями		
6	n	<NUM>	INT	Число отверстий		
7		<_VARI>	INT	зарезервировано		
8		<_UMODE>	INT	зарезервировано		
9		<_HIDE>	STRING [200]	Пропущенные позиции • макс. 198 знаков • Указание текущего номера позиции, к примеру, "1,3" (позиции 1 и 3 не выполняются)		
10		<_NSP>	INT	зарезервировано		
11		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/18/19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)

20.1.4 HOLES2 - Окружность отверстий

Синтаксис

```
HOLES2 (<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

Параметр

№	Параметр в маске	Параметр внутр.	Тип данных	Значение		
1	X0	<CPA>	REAL	Центр для окружности отверстий в 1-й оси (абс.) Исходная точка в 1-й оси	(при XY) (при XA, YB, ZC)	
2	Y0	<CPO>	REAL	Центр для окружности отверстий в 2-й оси (абс.) Исходная точка во 2-й оси	(при XY) (при XA, YB, ZC)	
3	r	<RAD>	REAL	Радиус окружности отверстий	(при XY)	
4	$\alpha 0$	<STA1>	REAL	Начальный угол или 1-я позиция круговой оси	(при XY) (при XA, YB, ZC)	
5	$\alpha 1$	<INDA>	REAL	Угол повторного включения (только для дели- тельной окружности)	(при XY, XA, YB, ZC)	
					< 0 =	по часовой стрелке
					> 0 =	против часовой стрелки
6	n	<NUM>	INT	Число позиций		

№	Параметр в маске	Параметр внутр.	Тип данных	Значение		
7		<_VARI>	INT	Режим обработки		
				ПЕРВАЯ ПОЗИЦИЯ:	зарезервировано	
				ВТОРАЯ ПОЗИЦИЯ:	тип позиционирования	
					0 =	подвод в позицию - линейный
					1 =	подвод в позицию - по круговой траектории
				ТРЕТЬЯ ПОЗИЦИЯ:	зарезервировано	
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	Образец окружности	
					0 =	режим совместимости, если INDA = 0 тогда полный круг, INDA <> 0 тогда делительная окружность)
					1 =	Полный круг
					2 =	Делительная окружность
				ПЯТАЯ ПОЗИЦИЯ:	образец позиции с круговой осью	
					0 =	XY (без круговой оси) (при XY)
					1 =	XA (ось X и круговая ось вокруг X) (только при XA)
					2 =	YB (ось Y и круговая ось вокруг Y) (только при YB)
					3 =	ZC (ось Z и круговая ось вокруг C) (только при ZC)
				СЕДЬМАЯ ПОЗИЦИЯ + ШЕСТАЯ ПОЗИЦИЯ:	смещение (при нескольких круговых осях вокруг одной и той же оси; если индекс слишком велик, то 1-й оси)	
					00 =	1-я ось A, B или C
					01 =	2-я ось A, B или C
					...	
					10 =	20-я ось A, B или C
8		<_UMODE>	INT	зарезервировано		
9		<_HIDE>	STRING [200]	зарезервировано		
10		<_NSP>	INT	зарезервировано		
11		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/18/19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)

20.1.5 РОCKET3 - Фрезерование прямоугольного кармана

Синтаксис

```
POCKET3(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_CRAD>,
<_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>,
<_CDIR>, <_VARI>, <_MIDA>, <_AP1>, <_AP2>, <_AD>, <_RAD1>, <_DP1>,
<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение		
1	RP	<_RTP>	REAL	Плоскость отвода (абс.)		
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)		
3	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)		
4	Z1	<_DP>	REAL	Глубина кармана (абс./инкр.), см. <_AMODE>)		
5	l	<_LENG>	REAL	Длина кармана (инкр., ввод со знаком)		
6	W	<_WID>	REAL	Ширина кармана (инкр., ввод со знаком)		
7	R	<_CRAD>	REAL	Радиус угла кармана		
8	X0	<_PA>	REAL	Исходная точка, 1-я ось (абс.)		
9	YO	<_PO>	REAL	Исходная точка, 2-я ось (абс.)		
10	α0	<_STA>	REAL	Угол поворота, угол между продольной осью (L) и 1-й осью		
11	DZ	<_MID>	REAL	Макс. подача на глубину		
12	UXY	<_FAL>	REAL	Чистовой припуск плоскость		
13	UZ	<_FALD>	REAL	Чистовой припуск глубина		
14	F	<_FFP1>	REAL	Подача в плоскости		
15	FZ	<_FFD>	REAL	Подача на глубину		
16		<_CDIR>	INT	Направление фрезерования:	0 =	Синхронный ход
					1 =	Противоход
17		<_VARI>	INT	Режим обработки		
				ПЕРВАЯ ПОЗИЦИЯ:		
					1 =	Черновая обработка
					2 =	Чистовая обработка
					4 =	Чистовая обработка края
					5 =	Снятие фаски
				ВТОРАЯ ПОЗИЦИЯ:		
					0 =	предварительное сверление, подача с G0
					1 =	вертикально, подача с G1
					2 =	по спирали
					3 =	качанием на продольной оси кармана
				ТРЕТЬЯ ПОЗИЦИЯ:	зарезервировано	

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
18	DXY	<_MIDA>	REAL	Макс. подача в плоскости, единица см. <_AMODE>
19	L1	<_AP1>	REAL	Длина предварительной обработки (инкр.)
20	W1	<_AP2>	REAL	Ширина предварительной обработки (инкр.)
21	AZ	<_AD>	REAL	Глубина предварительной обработки (инкр.)
22	ER	<_RAD1>	REAL	Радиус спиральной траектории при врезании по спирали
	EW			Макс. угол врезания для качания
23	EP	<_DP1>	REAL	Шаг спирали при врезании по спирали
24		<_UMODE>	INT	зарезервировано
25	FS	<_FS>	REAL	Ширина фаски (инкр.)
26	ZFS	<_ZFS>	REAL	Глубина врезания (острие инструмента) при снятии фаски (абс./инкр.) см. <_AMODE>
27		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)
				ПЕРВАЯ ПОЗИЦИЯ:
				зарезервировано
				ВТОРАЯ ПОЗИЦИЯ:
				зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:
				выбор обработки/только расчет стартовой точки
				0 =
				Режим совместимости
				1 =
				обычная обработка
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				указание размеров через центр/угол
				0 =
				Режим совместимости
				1 =
				указание размеров через центр
				2 =
				указание размеров угловая точка, длина кармана +LENG/+WID
				3 =
				указание размеров угловая точка, длина кармана -LENG/+WID
				4 =
				указание размеров угловая точка, длина кармана +LENG/-WID
				5 =
				указание размеров угловая точка, длина кармана -LENG/-WID
				ПЯТАЯ ПОЗИЦИЯ:
				полная обработка/доводка
				0 =
				режим совместимости (<_AP1>, <_AP2> и <_AD> обрабатывать как прежде)
				1 =
				Комплексная обработка
				2 =
				Доработка

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
28		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:
				тип подачи: G-группа (G94/G95) для подачи в плоскости и на глубину
				0 = Режим совместимости
29		<_AMODE>	INT	1 = G-команда как перед вызовом цикла. G94/G95 для подачи в плоскости и на глубину одновременно
				ТРЕТЬЯ ПОЗИЦИЯ: --- зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: --- зарезервировано
				ПЯТАЯ ПОЗИЦИЯ:
				Технологическое масштабирование внутри масок цикла (Страница 870)
				0 = Ввод: полный
				1 = Ввод: простой
				Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				глубина кармана (Z1)
				0 = абсолютная (режим совместимости)
				1 = инкрементальная
				ВТОРАЯ ПОЗИЦИЯ:
				единица для подачи в плоскости (DXY)
				0 = мм
				1 = % от диаметра инструмента
				ТРЕТЬЯ ПОЗИЦИЯ:
				глубина врезания при снятии фаски (ZFS)
				0 = абсолютная
				1 = инкрементальная

20.1.6 РОCKET4 - Фрезерование кругового кармана

Синтаксис

```
POCKET4(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_CDIAM>, <_PA>, <_PO>,
<_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>, <_MIDA>,
<_AP1>, <_AD>, <_RAD1>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
1	RP	<_RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)
3	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
4	Z1	<_DP>	REAL	Глубина кармана (абс./инкр.), см. <_AMODE>
5	Ø	<_CDIAM>	REAL	Диаметр кармана или радиус кармана, см. <_DMODE>
6	X0	<_PA>	REAL	Исходная точка 1-й оси (абс.)
7	Y0	<_PO>	REAL	Исходная точка 2-й оси (абс.)
8	DZ	<_MID>	REAL	Макс. подача на глубину, см. <_VARI>= в плоскости Макс. шаг спирали, см. <_VARI>= по спирали
9	UXY	<_FAL>	REAL	Чистовой припуск плоскость
10	UZ	<_FALD>	REAL	Чистовой припуск глубина
11	F	<_FFP1>	REAL	Подача для обработки поверхностей
12	FZ	<_FFD>	REAL	Подача на глубину
13		<_CDIR>	INT	Направление фрезерования 0 = Синхронный ход 1 = Противоход
14		<_VARI>	INT	Режим обработки ПЕРВАЯ ПОЗИЦИЯ: Обработка 1 = Черновая обработка 2 = чистовая обработка 4 = Чистовая обработка края 5 = Снятие фаски ВТОРАЯ ПОЗИЦИЯ: тип подачи (черновая и чистовая обработка) 0 = предварительное сверление, подача с G0 (карман предварительно обработан) 1 = вертикально, подача с G1 2 = по спирали ТРЕТЬЯ ПОЗИЦИЯ: зарезервировано ЧЕТВЕРТАЯ ПОЗИЦИЯ: 0 = в плоскости 1 = по спирали
15	DXY	<_MIDA>	REAL	Макс. подача в плоскости, см. <_AMODE>, 0 = 0,8 x диаметр инструмента
16	Ø	<_AP1>	REAL	Диаметр/радиус предварительной обработки (инкр.)
17	AZ	<_AD>	REAL	Глубина предварительной обработки (инкр.)
18	ER	<_RAD1>	REAL	Радиус спиральной траектории при врезании по спирали
19	EP	<_DP1>	REAL	Шаг спирали при врезании по спиральной траектории
20		<_UMODE>	INT	зарезервировано
21	FS	<_FS>	REAL	Ширина фаски (инкр.)

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
22	ZFS	<_ZFS>	REAL	Глубина врезания (острие инструмента) при снятии фаски (абс./инкр.) см. <_AMODE>
23		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений) ПЕРВАЯ ПОЗИЦИЯ: зарезервировано ВТОРАЯ ПОЗИЦИЯ: зарезервировано ТРЕТЬЯ ПОЗИЦИЯ: обработка/расчет стартовой точки 0 = Режим совместимости 1 = обычная обработка ЧЕТВЕРТАЯ ПОЗИЦИЯ: зарезервировано ПЯТАЯ ПОЗИЦИЯ: полная обработка/доводка 0 = режим совместимости (<_AP1> и <_AD> обрабатывать как прежде) 1 = Комплексная обработка 2 = Доработка

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
25		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				глубина кармана (Z1)
				0 = абсолютная (режим совместимости)
				1 = инкрементальная
				ВТОРАЯ ПОЗИЦИЯ:
				единица для ширины подачи (DXY)
				0 = мм
				1 = % от диаметра инструмента
				ТРЕТЬЯ ПОЗИЦИЯ:
				глубина врезания при снятии фаски (ZFS)
				0 = абсолютная
				1 = инкрементальная

20.1.7 SLOT1 - Продольный паз

Синтаксис

SLOT1 (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <WID>, <_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>, <CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FALD>, <_STA2>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<RFP>	REAL	Исходная точка оси инструмента (абс.)
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
4	Z1	<_DP>	REAL	Глубина паза (абс.)
5		<_DPR>	REAL	Глубина паза (инкр.), относительно Z0 (вводится без знака)
6		<NUM>	INT	Число пазов = 1
7	I	<LENG>	REAL	Длина паза
8	W	<WID>	REAL	Ширина паза
9	X0	<_CPA>	REAL	Исходная точка, 1-я ось плоскости
10	Y0	<_CPO>	REAL	Исходная точка, 2-я ось плоскости
11		<RAD>	REAL	зарезервировано
12	α	<STA1>	REAL	Угол поворота
13		<INDA>	REAL	зарезервировано
14	FZ	<FFD>	REAL	Подача на глубину
15	F	<FFP1>	REAL	Подача

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
16	DZ	<_MID>	REAL	Макс. подача на глубину	
17		<CDIR>	INT	Направление фрезерования	0 = Синхронный ход
					1 = Противоход
18	UXY	<_FAL>	REAL	Чистовой припуск Плоскость или край паза	
19		<VARI>	INT	Режим обработки	
				ПЕРВАЯ ПОЗИЦИЯ:	
					0 = зарезервировано
					1 = Черновая обработка
					2 = Чистовая обработка
					4 = чистовая обработка края (только обработка края)
					5 = Снятие фаски
				ВТОРАЯ ПОЗИЦИЯ:	подвод
					0 = предварительное сверление, подача с G0 (паз подготовлен)
					1 = вертикально, подача с G1
					2 = по спирали
					3 = качанием
				ТРЕТЬЯ ПОЗИЦИЯ:	зарезервировано
20	DZF	<_MIDF>	REAL	зарезервировано	
21	FF	<FFP2>	REAL	зарезервировано	
22	SF	<SSF>	REAL	зарезервировано	
23	UZ	<_FALD>	REAL	Чистовой припуск глубина	
24	ER	<_STA2>	REAL	Радиус спиральной траектории при врезании по спирали	
	EW			Макс. угол врезания для качания	
25	EP	<_DP1>	REAL	Глубина врезания на оборот для спирали	
26		<_UMODE>	INT	зарезервировано	
27	FS	<_FS>	REAL	Ширина фаски (инкр.) при снятии фаски	
28	ZFS	<_ZFS>	REAL	Глубина врезания (острие инструмента) при снятии фаски (абс./инкр.) см. <_AMODE>	

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
29		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: выбор обработки или только расчет стартовой точки
				1 = обычная обработка
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: указание размеров исходной точки, положения паза
				0 = центр
				1 = слева внутри +L
				2 = справа внутри -L
				3 = левый край +L
				4 = правый край -L
30		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ: плоскость обработки G17/18/19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: идентификатор версии ПО
				1 = расширение функций SLOT1
31		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ: конечная глубина Z1 (абс./инкр.)
				0 = Совместимость
				1 = Z1 (инкр.)
				2 = Z1 (абс.)
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: глубина врезания при снятии фаски ZFS
				0 = ZFS (абс.)
				1 = ZFS (инкр.)

Примечание

Цикл, по сравнению с предшествующими версиями ПО, получил новые функции. Следствием этого явилось то, что определенные параметры более не отображаются в маске ввода (<NUM>, <RAD>, <INDA>). Несколько пазов на одном образце позиций могут быть запрограммированы посредством "MCALL" и вызова требуемого образца позиций, к примеру, HOLES2.

20.1.8 SLOT2 - кольцевая канавка**Синтаксис**

SLOT2 (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <AFSL>, <WID>, <_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>, <CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FFCP>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
1	RP	<RTP>	REAL	Плоскость отвода (абс.)		
2	Z0	<RFP>	REAL	Исходная точка оси инструмента (абс.)		
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)		
4	Z1	<_DP>	REAL	Глубина паза (абс.)		
5		<_DPR>	REAL	Глубина паза (инкр.), относительно Z0 (вводится без знака)		
6	n	<NUM>	INT	Количество канавок		
7	$\alpha 1$	<AFSL>	REAL	Апертурный угол канавки		
8	W	<WID>	REAL	Ширина паза		
9	X0	<_CPA>	REAL	Исходная точка = центр окружности, 1-я ось плоскости		
10	Y0	<_CPO>	REAL	Исходная точка = центр окружности, 2-я ось плоскости		
11	r	<RAD>	REAL	Радиус окружности		
12	$\alpha 0$	<STA1>	REAL	Начальный угол		
13	$\alpha 2$	<INDA>	REAL	Угол повторного включения		
14	FZ	<FFD>	REAL	Подача на глубину		
15	F	<FFP1>	REAL	Подача		
16	DZ	<_MID>	REAL	Макс. подача на глубину		
17		<CDIR>	INT	Направление фрезерования	0 =	Синхронный ход
					1 =	Противоход
18	UXY	<_FAL>	REAL	Чистовой припуск Плоскость или край паза		

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
19		<VARI>	INT	Режим обработки
				ПЕРВАЯ ПОЗИЦИЯ:
				0 = Комплексная обработка
				1 = Черновая обработка
				2 = Чистовая обработка
				3 = Чистовая обработка края
				5 = Снятие фаски
				ВТОРАЯ ПОЗИЦИЯ:
				0 = промежуточное позиционирование с G0-прямой
				1 = промежуточное позиционирование по круговой траектории
				ТРЕТЬЯ ПОЗИЦИЯ:
				зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				0 = режим совместимости, если <INDA> = 0, то полный круг, <INDA><> 0, то часть круга)
				1 = Полный круг
				2 = Делительная окружность
20	DZF	<_MIDF>	REAL	зарезервировано
21		<FFP2>	REAL	зарезервировано
22		<SSF>	REAL	зарезервировано
23	FF	<_FFCP>	REAL	зарезервировано
24		<_UMODE>	INT	зарезервировано
25	FS	<_FS>	REAL	Ширина фаски (инкр.)
26	ZFS	<_ZFS>	REAL	Глубина врезания (острие инструмента) при снятии фаски (абс./инкр.) см. <_AMODE>
27		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)
				ПЕРВАЯ ПОЗИЦИЯ:
				зарезервировано
				ВТОРАЯ ПОЗИЦИЯ:
				зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:
				выбор обработки или только расчет стартовой точки
				0 = Режим совместимости
				1 = обычная обработка

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
28		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				плоскость обработки G17/18/19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:
				зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:
29		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				конечная глубина Z1 (абс./инкр.)
				0 = Совместимость
				1 = Z1 (инкр.)
				2 = Z1 (абс.)
				ВТОРАЯ ПОЗИЦИЯ:
				зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:
				глубина врезания при снятии фаски ZFS

20.1.9 LONGHOLE - Продольный паз

Синтаксис

```
LONGHOLE (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <_CPA>,
<_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>, <_VARI>,
<_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
1	RP	<RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
4	Z1	<_DP>	REAL	Глубина продольного паза (абс.)
5		<_DPR>	REAL	Глубина продольного паза (инкр.), относительно Z0 (вводится без знака)
6		<NUM>	INT	Число продольных пазов = 1
7	I	<LENG>	REAL	Длина продольного паза
8	X0	<_CPA>	REAL	Исходная точка, 1-я ось плоскости
9	Y0	<_CPO>	REAL	Исходная точка, 2-я ось плоскости
10		<RAD>	REAL	зарезервировано
11	$\alpha 0$	<STA1>	REAL	Угол поворота
12		<INDA>	REAL	зарезервировано
13	FZ	<FFD>	REAL	Подача на глубину
14	F	<FFP1>	REAL	Подача
15	DZ	<MID>	REAL	Макс. подача на глубину
16		<_VARI>	INT	Режим обработки
				ПЕРВАЯ ПОЗИЦИЯ:
				тип подачи
				1 = вертикально с G1
				3 = качанием
				ТРЕТЬЯ ПОЗИЦИЯ:
				зарезервировано
17		<_UMODE>	INT	зарезервировано
18		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)
				ПЕРВАЯ ПОЗИЦИЯ:
				зарезервировано
				ВТОРАЯ ПОЗИЦИЯ:
				зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:
				выбор обработки или только расчет стартовой точки
				0 = режим совместимости
				1 = обычная обработка
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				указание размеров исходной точки, положения паза
				0 = центр
				1 = слева внутри +L
				2 = справа внутри -L
				3 = левый край +L
				4 = правый край -L

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
19		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				плоскость обработки G17/18/19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:
20		<_AMODE>	INT	тип подачи: G-группа (G94/G95) для подачи в плоскости и на глубину
				0 = режим совместимости
				1 = G-команда как перед вызовом цикла. G94/G95 для подачи в плоскости и на глубину одновременно
				ТРЕТЬЯ ПОЗИЦИЯ:
				зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				идентификатор версии ПО
				1 = расширение функций LONGHOLE (указание размеров исходной точки)
		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				конечная глубина Z1 (абс./инкр.)
				0 = Совместимость
				1 = Z1 (инкр.)
				2 = Z1 (абс.)

Примечание

Цикл, по сравнению с предшествующими версиями ПО, получил новые функции. Следствием этого явилось то, что определенные параметры более не отображаются в маске ввода (<NUM>, <RAD>, <INDA>). Несколько пазов на одном образце позиций могут быть запрограммированы посредством "MCALL" и вызова требуемого образца позиций, например, HOLES2.

20.1.10 CYCLE60 - Цикл гравирования**Синтаксис**

```
CYCLE60 (<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
1		<_TEXT>	STRING [200]	Гравируемый текст (макс. 100 символов)
2	RP	<_RTP>	REAL	Плоскость отвода (абс.)
3	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)
4	sc	<_SDIS>	REAL	Безопасное расстояние (аддитивно к базовой плоскости, вводится без знака)
5	Z1	<_DP>	REAL	Глубина (абс.), см. <_AMODE>
6	Z1	<_DPR>	REAL	Глубина (инкр.), см. <_AMODE>
7	X0	<_PA>	REAL	Исходная точка, 1-я ось плоскости (абс.) - прямоуг., см. <_VARI>
	R			Исходная точка, длина (радиус) - поляр., см. <_VARI>
8	Y0	<_PO>	REAL	Исходная точка, 2-я ось плоскости (абс.) - прямоуг., см. <_VARI>
	$\alpha 0$			Исходная точка, угол относительно 1-ой оси — полярн., см. <_VARI>
9	$\alpha 1$	<_STA>	REAL	Направление текста, угол текстовой линии относительно 1-ой оси), см. <_VARI>
10	XM	<_CP1>	REAL	Центр текстовой окружности, 1-я ось плоскости (абс.) - прямоуг., см. <_VARI>
	LM			Центр текстовой окружности, длина (радиус) относительно WNP - полярн., см. <_VARI>
11	YM	<_CP2>	REAL	Центр текстовой окружности, 2-я ось плоскости (абс.) - прямоуг., см. <_VARI>
	αM			Центр текстовой окружности, угол относительно 1-ой оси — полярн., см. <_VARI>
12	W	<_WID>	REAL	Высота символов (вводится без знака)
13	DX1	<_DF>	REAL	Интервал между символами / общая ширина, см. <_VARI>
	DX2			
	$\alpha 2$			Апертурный угол, см. <_VARI>
14	FZ	<_FFD>	REAL	Подача на глубину, см. <_DMODE>
15	F	<_FFP1>	REAL	Подача для обработки поверхностей

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
16		<_VARI>	INT	Обработка (ориентация и исходная точка гравированного текста)
				ПЕРВАЯ ПОЗИЦИЯ:
				исходная точка
				0 = прямоуг.
				1 = поляр.
				ВТОРАЯ ПОЗИЦИЯ:
				ориентация текста
				0 = текст на одной линии
				1 = текст на дуге окружности сверху
				2 = текст на дуге окружности снизу
				ТРЕТЬЯ ПОЗИЦИЯ:
				зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				исходная точка текста горизонтально
				0 = слева
				1 = по центру
				2 = справа
17		<_CODEP>	INT	исходная точка текста вертикально
				0 = внизу
				1 = по центру
				2 = сверху
				ШЕСТАЯ ПОЗИЦИЯ:
				длина текста
				0 = интервал между символами
				1 = общая ширина текста (только для линейного текста)
				2 = апертурный угол (только для текста на дуге окружности)
				СЕДЬМАЯ ПОЗИЦИЯ:
				центр окружности
				0 = прямоугольная (декартова)
				1 = полярная
				ВОСЬМАЯ ПОЗИЦИЯ:
				зеркальный шрифт
				0 = Совместимость
				1 = зеркальный шрифт ВКЛ
				2 = зеркальный шрифт ВЫКЛ
18		<_UMODE>	INT	зарезервировано
19		_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)
				ПЕРВАЯ ПОЗИЦИЯ:
				зарезервировано
				ВТОРАЯ ПОЗИЦИЯ:
				зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:
				выбор обработки/только расчет стартовой точки
				0 = Режим совместимости
				1 = обычная обработка

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
20		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				плоскость обработки G17/18/19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:
21		<_AMODE>	INT	тип подачи: G-группа (G94/G95) для подачи в плоскости и на глубину
				0 = Режим совместимости
				1 = G-команда как перед вызовом цикла. G94/G95 для подачи в плоскости и на глубину одновременно
				Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				конечная глубина (<_DP>, <_DPR>)
				0 = Совместимость
				1 = инкрементальная (<_DPR>)
				2 = абсолютная (<_DP>)

20.1.11 CYCLE61 - плоское фрезерование

Синтаксис

```
CYCLE61(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<_RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента, высота заготовки (абс.)
3	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
4	Z1	<_DP>	REAL	Высота готовой детали (абс./инкр.), см. <_AMODE>
5	X0	<_PA>	REAL	Угловая точка 1 в 1-й оси (абс.)
6	Y0	<_PO>	REAL	Угловая точка 1 во 2-й оси (абс.)
7	X1	<_LENG>	REAL	Угловая точка 2 в 1-й оси (абс./инкр.), см. <_AMODE>
8	Y1	<_WID>	REAL	Угловая точка 2 во 2-й оси (абс./инкр.), см. <_AMODE>
9	DZ	<_MID>	REAL	Макс. подача на глубину

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
10	DXY	<_MIDA>	REAL	Макс. подача в плоскости (единица, см. <_AMODE>)	
11	UZ	<_FALD>	REAL	Чистовой припуск глубина	
12	F	<_FFP1>	REAL	Подача обработки	
13		<_VARI>	INT	Режим обработки	
				ПЕРВАЯ ПОЗИЦИЯ:	обработка
					1 = Черновая обработка
					2 = Чистовая обработка
				ВТОРАЯ ПОЗИЦИЯ:	направление обработки
					1 = параллельно 1-й оси, одно направление
					2 = параллельно 2-й оси, одно направление
					3 = параллельно 1-й оси, разное направление
					4 = параллельно 2-й оси, разное направление
14		<_LIM>	INT	Ограничения	
				ПЕРВАЯ ПОЗИЦИЯ:	ограничение 1-й оси минус
					0 = нет
					1 = да
				ВТОРАЯ ПОЗИЦИЯ:	ограничение 1-й оси плюс
					0 = нет
					1 = да
				ТРЕТЬЯ ПОЗИЦИЯ:	ограничение 2-й оси минус
					0 = нет
					1 = да
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	ограничение 2-й оси плюс
					0 = нет
					1 = да
15		<_DMODE>	INT	Режим индикации	
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/18/19
					0 = совместимость, остается активной действующая перед вызовом цикла плоскость
					1 = G17 (активна только в цикле)
					2 = G18 (активна только в цикле)
					3 = G19 (активна только в цикле)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
16		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				конечная глубина (<_DP>)
				0 = абсолютная
				1 = инкрементальная
				ВТОРАЯ ПОЗИЦИЯ:
				единица для подачи в плоскости (<_MIDA>)
				0 = мм
				1 = % от диаметра инструмента
				ТРЕТЬЯ ПОЗИЦИЯ: зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				длина плоскости
				0 = инкрементальная
				1 = абсолютная
				ПЯТАЯ ПОЗИЦИЯ:
				ширина плоскости
				0 = инкрементальная
				1 = абсолютная

20.1.12 CYCLE62 - Вызов контура

Синтаксис

CYCLE62 (<_KNAME>, <_TYPE>, <_LAB1>, <_LAB2>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	PRG/CON	<_KNAME>	STRING [140]	Имя контура или имя подпрограммы, программирования не требуется при _TYPE = 2
2		<_TYPE>	INT	Определение ввода контура
				0 = Подпрограмма
				1 = Имя контура
				2 = Метки
				3 = Метки в подпрограмме
3	LAB1	<_LAB1>	STRING[32]	Метка 1, начало контура
4	LAB2	<_LAB2>	STRING[32]	Метка 2, конец контура

20.1.13 CYCLE63 - фрезерование кармана

Синтаксис

```
CYCLE63(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_FZ>,
<_DXY>, <_DZ>, <_UXY>, <_UZ>, <_CDIR>, <_XS>, <_YS>, <_ER>, <_EP>,
<_EW>, <_FS>, <_ZFS>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	PRG	<_PRG>	STRING [100]	Имя программы выборки
2		<_VARI>	INT	Режим обработки ПЕРВАЯ ПОЗИЦИЯ: технологическая обработка 1 = Черновая обработка 3 = Чистовая обработка основания 4 = Чистовая обработка края 5 = Снятие фаски ВТОРАЯ ПОЗИЦИЯ: тип подачи 0 = Врезание по центру 1 = Спиральное врезание 2 = Маятниковое врезание ТРЕТЬЯ ПОЗИЦИЯ: зарезервировано ЧЕТВЕРТАЯ ПОЗИЦИЯ: режим подъема 0 = подъем на плоскость отвода 1 = подъем на исходную точку + безопасное расстояние ПЯТАЯ ПОЗИЦИЯ: стартовая точка при черновой и чистовой обработке основания 0 = автоматически 1 = вручную
3	RP	<_RP>	REAL	Плоскость отвода (абс.)
4	Z0	<_Z0>	REAL	Исходная точка оси инструмента (абс.)
5	sc	<_SC>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
6	Z1	<_Z1>	REAL	Конечная глубина, (см. <_AMODE> ПЕРВАЯ ПОЗИЦИЯ)
7	F	<_F>	REAL	Подача в плоскости черновая/чистовая обработка
8	FZ	<_FZ>	REAL	Подача на глубину
9	DXY	<_DXY>	REAL	Подача в плоскости - единица, (см. <_AMODE> ВТОРАЯ ПОЗИЦИЯ)
10	DZ	<_DZ>	REAL	Подача на глубину
11	UXY	<_UXY>	REAL	Чистовой припуск плоскость
12	UZ	<_UZ>	REAL	Чистовой припуск глубина

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
13		<_CDIR>	INT	Направление фрезерования	0 =	Синхронный ход
					1 =	Противоход
14	XS	<_XS>	REAL	Стартовая точка X, абсолютная		
15	YS	<_YS>	REAL	Стартовая точка Y, абсолютная		
16	ER	<_ER>	REAL	Врезание по спирали: Радиус		
17	EP	<_EP>	REAL	Врезание по спирали: Шаг		
18	EW	<_EW>	REAL	Маятниковое врезание: Макс. угол врезания		
19	FS	<_FS>	REAL	Ширина фаски (инкр.) при снятии фаски		
20	ZFS	<_ZFS>	REAL	Глубина врезания острия инструмента при снятии фаски, (см. <_AMODE>ТРЕТЬЯ ПОЗИЦИЯ)		
21	tr	<_TR>	STRING[32]	Имя эталонного инструмента при обработке оставшегося материала		
22	DR	<_DR>	INT	D-номер эталонного инструмента при обработке оставшегося материала		
23		<_UMODE>	INT	зарезервировано		
24		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)		
				ПЕРВАЯ ПОЗИЦИЯ:	зарезервировано	
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано	
				ТРЕТЬЯ ПОЗИЦИЯ:	выбор обработки/только расчет стартовой точки	
					0 =	обычная обработка (режима совместимости не требуется)
					1 =	обычная обработка
					2 =	зарезервировано
25		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано	
				ТРЕТЬЯ ПОЗИЦИЯ:	технологический режим	
					1 =	карман
					2 =	цапфа
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	обработка остаточного материала	
					0 =	нет
					1 =	да
				ПЯТАЯ ПОЗИЦИЯ:	Технологическое масштабирование внутри масок цикла (Страница 870)	
					0 =	Ввод: полный
					1 =	Ввод: простой

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
26		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				конечная глубина (Z1)
				0 = абсолютная (режим совместимости)
				1 = инкрементальная
				ВТОРАЯ ПОЗИЦИЯ:
				единица для подачи в плоскости (DXY)
				0 = мм
				1 = % от диаметра инструмента
				ТРЕТЬЯ ПОЗИЦИЯ:
				глубина врезания при снятии фаски (ZFS)
				0 = абсолютная
				1 = инкрементальная
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: --- зарезервировано

20.1.14 CYCLE64 - Предварительное сверление кармана

Синтаксис

```
CYCLE64(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_DXY>,
<_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	PRG	<_PRG>	STRING [100]	Имя программы сверления/центрования
2		<_VARI>	INT	Режим обработки
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: режим подъема
				0 = подъем на плоскость отвода
				1 = подъем на исходную точку + безопасное расстояние
3	RP	<_RP>	REAL	Плоскость отвода (абс.)
4	Z0	<_Z0>	REAL	Исходная точка (абс.)
5	sc	<_SC>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
6	Z1	<_Z1>	REAL	Глубина сверления/центрования, (см. <_AMODE>ПЕРВАЯ ПОЗИЦИЯ)
7	F	<_F>	REAL	Подача сверления/центрования

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
8	DXY	<_DXY>	REAL	Подача в плоскости - единица, (см. <_AMODE>ВТОРАЯ ПОЗИЦИЯ)		
9	UXY	<_UXY>	REAL	Чистовой припуск плоскость		
10	UZ	<_UZ>	REAL	Чистовой припуск глубина		
11		<_CDIR>	INT	Направление фрезерования	0 =	Синхронный ход
					1 =	Противоход
12	tr	<_TR>	STRING[20]	Имя эталонного инструмента		
13	DR	<_DR>	INT	D-номер эталонного инструмента		
14		<_UMODE>	INT	зарезервировано		
15		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)		
				ПЕРВАЯ ПОЗИЦИЯ:	зарезервировано	
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано	
				ТРЕТЬЯ ПОЗИЦИЯ:	выбор обработки/только расчет стартовой точки	
					0 =	обычная обработка (режима совместимости не требуется)
					1 =	обычная обработка
					2 =	зарезервировано
25		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:	технологический режим	
					1 =	Предварительное сверление
					2 =	Центрование
26		<_AMODE>	INT	Альтернативный режим		
				ПЕРВАЯ ПОЗИЦИЯ:	глубина сверления/центрования Z1	
					0 =	абсолютная (режим совместимости)
					1 =	инкрементальная
				ВТОРАЯ ПОЗИЦИЯ:	единица для подачи в плоскости (DXY)	
					0 =	мм
					1 =	% от диаметра инструмента

20.1.15 CYCLE70 - резьбофрезерование

Синтаксис

```
CYCLE70 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Параметры

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
1	RP	<_RTP>	REAL	Плоскость отвода (абс.)		
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)		
3	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)		
4	Z1	<_DP>	REAL	Длина резьбы (абс., инкр.), см. <_AMODE> Учитывать выход на основании отверстия (минимум половина шага)		
5	Ø	<_DIATH>	REAL	Ном. диаметр резьбы		
6	H1	<_H1>	REAL	Глубина резьбы		
7	U	<_FAL>	REAL	Чистовой припуск		
8	P	<_PIT>	REAL	Шаг резьбы (выбор <_PITA>: мм, дюйм, MODUL, витков/дюйм)		
9	NT	<_NT>	INT	Число зубьев на режущей пластине Длина инструмента всегда относится к нижнему зубу!		
10	DXY	<_MID>	REAL	Макс. подача на проход резца <_MID> > <_H1>: все за один проход резца		
11	F	<_FFR>	REAL	Подача фрезерования		
12		<_TYPTH>	INT	Тип резьбы	0 =	внутренняя резьба
					1 =	наружная резьба
13	X0	<_PA>	REAL	Центр окружности, 1-я ось (абс.)		
14	Y0	<_PO>	REAL	Центр окружности, 2-я ось (абс.)		
15	αS	<_NSP>	REAL	Начальный угол (многозаходная резьба)		
16		<_VARI>	INT	Режим обработки		
				ПЕРВАЯ ПОЗИЦИЯ:		
					1 =	черновая обработка
					2 =	чистовая обработка
				ВТОРАЯ ПОЗИЦИЯ:		
					1 =	сверху вниз
					2 =	снизу вверх
				ТРЕТЬЯ ПОЗИЦИЯ:		
					0 =	правая резьба
					1 =	левая резьба

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
17		<_PITA>	INT	Нормирование шага резьбы
				0 = режим совместимости
				1 = Шаг в мм
				2 = шаг в витках на дюйм (TPI)
				3 = шаг в дюймах
				4 = шаг как MODUL
18		<_PITM>	STRING[15]	Строка как меркер для ввода шага резьбы (только для интерфейса)
19		<_PTAB>	STRING[20]	Строка для таблицы резьб ("", "ISO", "BSW", "BSP", "UNC") (только для интерфейса)
20		<_PTABA>	STRING[20]	Строка для выбора в таблице резьб (к примеру, "M 10", "M 12", ...) (только для интерфейса)
21		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: обработка/расчет стартовой точки
				0 = режим совместимости
				1 = обычная обработка
22		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ: плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
23		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ: Длина резьбы (<_DP>)
				0 = абсолютная
				1 = инкрементальная

20.1.16 CYCLE72 - фрезерование траектории

Синтаксис

```
CYCLE72 (<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение	
1		<_KNAME>	STRING [141]	Имя подпрограммы контура	
2	RP	<_RTP>	REAL	Плоскость отвода (абс.)	
3	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)	
4	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)	
5	Z1	<_DP>	REAL	Конечная тока, конечная глубина (абс./инкр.), см. <_AMODE>	
6	DZ	<_MID>	REAL	Макс. подача на глубину (инкр., вводится без знака)	
7	UXY	<_FAL>	REAL	Чистовой припуск плоскость (инкр.), припуск на контуре края	
8	UZ	<_FALD>	REAL	Чистовой припуск глубина (инкр.), припуск на основании (вводится без знака)	
9	FX	<_FFP1>	REAL	Подача на конуре	
10	FZ	<_FFD>	REAL	Подача на глубину (или подача в пространстве)	
11		<_VARI>	INT	Режим обработки	
				ПЕРВАЯ ПОЗИЦИЯ:	обработка
				1 =	черновая обработка
				2 =	чистовая обработка
				5 =	снятие фаски
				ВТОРАЯ ПОЗИЦИЯ:	
				0 =	промежуточные пути с G0
				1 =	промежуточные пути с G1
				ТРЕТЬЯ ПОЗИЦИЯ:	отвод на конце контура
				0 =	отвод на конце контура на исходную точку
				1 =	отвод на конце контура на исходную точку + <_SDIS>
				2 =	отвод на конце контура на <_SDIS>
				3 =	нет отвода на конце контура, подвод в следующую стартовую точку с подачей по контуру
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	зарезервировано
				ПЯТАЯ ПОЗИЦИЯ:	Обработка контура
				0 =	обработка контура вперед
				1 =	обработка контура назад
					Ограничения при назад:
					• макс. 170 элементов контура (вкл. фаски и закругления)
					• обрабатываются только значения в плоскости (X/Y) и F

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
12		<_RL>	INT	Направление обработки
				40 = по центру контура (G40, подвод и отвод: прямая или вертикально)
				41 = слева от контура (G41, подвод и отвод: прямая или окружность)
				42 = справа от контура (G42, подвод и отвод: прямая или окружность)
13		<_AS1>	INT	Движение подвода к контуру
				ПЕРВАЯ ПОЗИЦИЯ:
				1 = прямая
				2 = четверть круга
				3 = полукруг
				4 = вертикальный подвод и отвод
				ВТОРАЯ ПОЗИЦИЯ:
				0 = последнее движение, в плоскости
				1 = последнее движение, в пространстве
14	L1	<_LP1>	REAL	Путь подвода, или радиус подвода (инкр., вводится без знака)
15	FZ	<_FF3>	REAL	Подача для промежуточных путей (G94/G95 как на контуре)
16		<_AS2>	INT	Движение отвода от контура (не для вертикального подвода/отвода)
				ПЕРВАЯ ПОЗИЦИЯ:
				1 = прямая
				2 = четверть круга
				3 = полукруг
				ВТОРАЯ ПОЗИЦИЯ:
				0 = последнее движение, в плоскости
				1 = последнее движение, в пространстве
17	L2	<_LP2>	REAL	Путь отвода, или радиус отвода (инкр., вводится без знака)
18		<_UMODE>	INT	зарезервировано
19	FS	<_FS>	REAL	Ширина фаски (инкр.)
20	ZFS	<_ZFS>	REAL	Глубина врезания (острие инструмента) при снятии фаски (абс./инкр.) см. <_AMODE>

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
21		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)
				ПЕРВАЯ ПОЗИЦИЯ:
				ВТОРАЯ ПОЗИЦИЯ:
				ТРЕТЬЯ ПОЗИЦИЯ:
				0 =
				1 =
22		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				0 =
				1 =
				2 =
				3 =
				ВТОРАЯ ПОЗИЦИЯ:
				0 =
				1 =
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
23		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				0 =
				1 =
				ВТОРАЯ ПОЗИЦИЯ:
				0 =
				1 =
				ТРЕТЬЯ ПОЗИЦИЯ:
				0 =
				1 =

Примечание

Если следующие передаваемые параметры запрограммированы косвенно (как параметры), то обратный перевод маски ввода не выполняется:

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

20.1.17 CYCLE76 - фрезерование прямоуг. цапфы**Синтаксис**

CYCLE76(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_LENG>, <_WID>, <_CRAD>, <_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>, <_AP1>, <_AP2>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<_RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)
3	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
4	Z1	<_DP>	REAL	Глубина цапфы (абс.)
5		<_DPR>	REAL	Глубина цапфы (инкр.), относительно Z0 (вводится без знака)
6	I	<_LENG>	REAL	Ширина цапфы, см. <_GMODE>(вводится без знака)
7	W	<_WID>	REAL	Ширина цапфы, см. <_GMODE>(вводится без знака)
8	r	<_CRAD>	REAL	Радиус угла цапфы (вводится без знака)
9	X0	<_PA>	REAL	Исходная точка цапфы, 1-я ось плоскости (абс.)
10	Y0	<_PO>	REAL	Исходная точка цапфы, 2-я ось плоскости (абс.)
11	α0	<_STA>	REAL	Угол поворота, угол между продольной осью (L) и 1-й осью плоскости
12	DZ	<_MID>	REAL	Макс. подача на глубину (инкр., вводится без знака)
13	UXY	<_FAL>	REAL	Чистовой припуск плоскость (инкр.), припуск на контуре края
14	UZ	<_FALD>	REAL	Чистовой припуск глубина (инкр.), припуск на основании (вводится без знака)
15	FX	<_FFP1>	REAL	Подача на конуре
16	FZ	<_FFD>	REAL	Подача на глубину
17		<_CDIR>	INT	Направление фрезерования (вводится без знака)
				ПЕРВАЯ ПОЗИЦИЯ:
				0 = синхронный ход
				1 = противоход

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
18		<_VARI>	INT	Обработка	
				ПЕРВАЯ ПОЗИЦИЯ:	
				1 =	черновая обработка
				2 =	чистовая обработка
				5 =	снятие фаски
19	L1	<_AP1>	REAL	Длина цапфы-заготовки	
20	W1	<_AP2>	REAL	Ширина цапфы-заготовки	
21	FS	<_FS>	REAL	Ширина фаски (инкр.)	
22	ZFS	<_ZFS>	REAL	Глубина врезания (острие инструмента) при снятии фаски (абс., инкр.) см. <_AMODE>	
23		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)	
				ПЕРВАЯ ПОЗИЦИЯ:	зарезервировано
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:	выбор обработки или только расчет стартовой точки
					0 = режим совместимости
					1 = обычная обработка
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	указание размеров цапфы через центр или угол
					0 = режим совместимости
					1 = указание размеров через центр
					2 = указание размера через угловую точку, цапфа +L +W
					3 = указание размера через угловую точку, цапфа -L +W
					4 = указание размера через угловую точку, цапфа +L -W
					5 = указание размера через угловую точку, цапфа -L -W
				ПЯТАЯ ПОЗИЦИЯ:	полная обработка или доводка
					0 = режим совместимости
					1 = комплексная обработка
					2 = доработка

№	Параметр в маске	Параметр внутренний	Тип данных	Значение				
24		<_DMODE>	INT	Режим индикации				
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/18/19			
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость		
					1 =	G17 (активна только в цикле)		
					2 =	G18 (активна только в цикле)		
					3 =	G19 (активна только в цикле)		
				ВТОРАЯ ПОЗИЦИЯ:		---	зарезервировано	
				ТРЕТЬЯ ПОЗИЦИЯ:		---	зарезервировано	
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:		---	зарезервировано	
				ПЯТАЯ ПОЗИЦИЯ:	Технологическое масштабирование внутри масок цикла (Страница 870)			
0 =	ввод: полный							
1 =	ввод: простой							
25		<_AMODE>	INT	Альтернативный режим				
				ПЕРВАЯ ПОЗИЦИЯ:	конечная глубина Z1 (DP)			
					0 =	совместимость		
					1 =	инкрементальная		
					2 =	абсолютная		
				ВТОРАЯ ПОЗИЦИЯ:		зарезервировано		
				ТРЕТЬЯ ПОЗИЦИЯ:	глубина врезания при снятии фаски (ZFS)			
					0 =	абсолютная		
					1 =	инкрементальная		

20.1.18 CYCLE77 - фрезерование круговой цапфы

Синтаксис

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>,
<_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<_RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)
3	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
4	Z1	<_DP>	REAL	Глубина цапфы (абс.)
5		<_DPR>	REAL	Глубина цапфы (инкр.), относительно Z0 (вводится без знака)
6	Ø	<_CDIAM>	REAL	Диаметр цапфы (вводится без знака)
7	X0	<_PA>	REAL	Исходная точка цапфы, 1-я ось плоскости (абс.)
8	Y0	<_PO>	REAL	Исходная точка цапфы, 2-я ось плоскости (абс.)
9	DZ	<_MID>	REAL	Макс. подача на глубину (инкр., вводится без знака)
10	UXY	<_FAL>	REAL	Чистовой припуск плоскость (инкр.), припуск на контуре края
11	UZ	<_FALD>	REAL	Чистовой припуск глубина (инкр.), припуск на основании (вводится без знака)
12	FX	<_FFP1>	REAL	Подача на конуре
13	FZ	<_FFD>	REAL	Подача на глубину
14		<_CDIR>	INT	Направление фрезерования (вводится без знака)
				ПЕРВАЯ ПОЗИЦИЯ:
				0 = синхронный ход
				1 = противоход
15		<_VARI>	INT	Режим обработки
				ПЕРВАЯ ПОЗИЦИЯ: Обработка
				1 = черновая обработка до чистового припуска
				2 = чистовая обработка (припуск X/Y/Z = 0)
				5 = снятие фаски
16	Ø1	<_AP1>	REAL	Диаметр цапфы-заготовки
17	FS	<_FS>	REAL	Ширина фаски (инкр.)
18	ZFS	<_ZFS>	REAL	Глубина врезания (острие инструмента) при снятии фаски (абс./инкр.) см. <_AMODE>
19		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: выбор обработки/только расчет стартовой точки
				0 = режим совместимости
				1 = обычная обработка
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: зарезервировано
				ПЯТАЯ ПОЗИЦИЯ: полная обработка/доводка
				0 = режим совместимости (<_AP1> обрабатывать как прежде)
				1 = комплексная обработка
				2 = доработка

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
20		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				плоскость обработки G17/18/19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ: --- зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: --- зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: --- зарезервировано
21		<_AMODE>	INT	ПЯТАЯ ПОЗИЦИЯ:
				Технологическое масштабирование внутри масок цикла (Страница 870)
				0 = ввод: полный
				1 = ввод: простой
				Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				конечная глубина Z1 (DP)
				0 = абсолютная (режим совместимости)
				1 = инкрементальная
				2 = абсолютная
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:
				глубина врезания при снятии фаски (ZFS)
				0 = абсолютная
				1 = инкрементальная

20.1.19 CYCLE78 - Сверлильное резьбофрезерование

Синтаксис

```
CYCLE78(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_ADPR>, <_FDPR>, <_LDPR>,
<_DIAM>, <_PIT>, <_PITA>, <_DAM>, <_MDEP>, <_VARI>, <_CDIR>, <_GE>,
<_FFD>, <_FRDP>, <_FFR>, <_FFP2>, <_FFA>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<_RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение								
3	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)								
4	Z1	<_DP>	REAL	Конечная глубина сверления (абс./инкр.), см. <_AMODE>								
5		<_ADPR>	REAL	Глубина засверливания с уменьшенной подачей сверления (инкр.), действует при <_VARI> ПЯТАЯ ПОЗИЦИЯ								
6	D	<_FDPR>	REAL	Макс. подача на глубину (инкр.) D ≥ Z1 ⇒ подача на конечную глубину сверления D < Z1 ⇒ цикл глубокого сверления с несколькими подачами и удалением стружки								
7	ZR	<_LDPR>	REAL	Остаточная глубина сверления при сквозном сверлении (инкр.), с подачей FR								
8	Ø	<_DIAM>	REAL	Ном. диаметр резьбы								
9	P	<_PIT>	REAL	Шаг резьбы как числовое значение								
10		<_PITA>	INT	Нормирование шага резьбы P <table><tr><td>1 =</td><td>шаг в мм/об</td></tr><tr><td>2 =</td><td>шаг в витках/дюйм</td></tr><tr><td>3 =</td><td>шаг в дюймах/об</td></tr><tr><td>4 =</td><td>шаг как MODUL</td></tr></table>	1 =	шаг в мм/об	2 =	шаг в витках/дюйм	3 =	шаг в дюймах/об	4 =	шаг как MODUL
1 =	шаг в мм/об											
2 =	шаг в витках/дюйм											
3 =	шаг в дюймах/об											
4 =	шаг как MODUL											
11	DF	<_DAM>	REAL	Величина/процент для каждой следующей подачи (депрессия), см. <_AMODE>								
12	V1	<_MDEP>	REAL	Мин. подача (инкр.), действует только при депрессии								

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
13		<_VARI>	INT	Режим обработки	
				ПЕРВАЯ ПОЗИЦИЯ:	зарезервировано
				ВТОРАЯ ПОЗИЦИЯ:	Удаление стружки перед резьбофрезерованием
					0 = без удаления стружки перед резьбофрезерованием (действует только на конечной глубине сверления)
					1 = удаление стружки перед резьбофрезерованием (действует только на конечной глубине сверления)
				ТРЕТЬЯ ПОЗИЦИЯ:	Правая/левая резьба
					0 = правая резьба
					1 = левая резьба
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	остаточная глубина сверления с подачей сверления
					0 = нет остаточной глубины сверления с подачей сверления FR
					1 = остаточная глубина сверления с подачей сверления FR
14		<_CDIR>	INT	Направление фрезерования	Засверливание с уменьшенной подачей
					0 = нет засверливания с уменьшенной подачей
					1 = засверливание с уменьшенной подачей
					4 = засверливание с уменьшенной подачей Подача засверливания = 0,3 F1, если F1 < 0,15 мм/об Подача засверливания = 0,1 мм/об, если F1 ≥ 0,15 мм/об
15	Z2	<_GE>	REAL	Величина отвода перед резьбофрезерованием (инкр.)	
16	F1	<_FFD>	REAL	Подача сверления (мм/мин или дюймов/мин или мм/об)	
17	FR	<_FRDP>	REAL	Подача сверления для остаточной глубины сверления (мм/мин или мм/об)	
18	F2	<_FFR>	REAL	Подача для резьбофрезерования (мм/мин или мм/зуб)	
19	FS	<_FFP2>	REAL	Чистовая подача для <_CDIR> = 4 (мм/мин или мм/зуб)	

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
20		<_FFA>	INT	Нормирование подач
				ПЕРВАЯ ПОЗИЦИЯ: подача сверления F1
				ВТОРАЯ ПОЗИЦИЯ: подача сверления для остаточной глубины сверления FR
				ТРЕТЬЯ ПОЗИЦИЯ: подача для резьбофрезерование F2
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: чистовая подача FS
21		<_PITM>	STRING[15]	Строка как меркер для ввода шага резьбы (только для интерфейса) ¹⁾
22		<_PTAB>	STRING[20]	Строка для таблицы резьб ("", "ISO", "BSW", "BSP", "UNC") (только для интерфейса) ¹⁾
23		<_PTABA>	STRING[20]	Строка для выбора в таблице резьб (к примеру, "М 10", "М 12", ...) (только для интерфейса) ¹⁾
24		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений), зарезервировано
25		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ: плоскость обработки G17/18/19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
26		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ: глубина сверления = конечная глубина сверления Z1 абс./инкр.
				0 = абсолютная
				1 = инкрементальная
				ВТОРАЯ ПОЗИЦИЯ: величина/процент DF для каждой следующей подачи (дегрессия)
				0 = значение
				1 = процент (от 0.001 до 100 %)

Примечание

¹⁾Параметры 21, 22 и 23 используются только при выборе резьбы в таблицах резьб маски ввода. Доступ к таблицам резьб через определение цикла при выполнении цикла невозможен.

20.1.20 CYCLE79 - многогранник

Синтаксис

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
1	RP	<_RTP>	REAL	Плоскость отвода (абс.)		
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)		
3	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)		
4	Z1	<_DP>	REAL	Глубина многогранника (абс./инкр.), см. <_AMODE>		
5	n	<_NUM>	INT	Число граней (1...n)		
6	SW / L	<_SWL>	REAL	Размер под ключ или длина грани (в зависимости от <_VARI>) ("SW" для размера под ключ, "L" для длины грани) Размер под ключ только при четном числе граней и одной грани		
7	X0	<_PA>	REAL	Исходная точка цапфы, 1-я ось (абс.)		
8	Y0	<_PO>	REAL	Исходная точка цапфы, 2-я ось (абс.)		
9	$\alpha 0$	<_STA>	REAL	Угол поворота центра грани к 1-й оси (ось X)		
10	R1/FS1	<_RC>	REAL	Закругление углов при <_NUM>> 2 (радиус/фаска, см. <_AMODE>) (инкр., ввод без знака) ("R1" для радиуса, "FS1" для фаски)		
11	Ø	<_AP1>	REAL	Диаметр трубы цапфы		
12	DXY	<_MIDA>	REAL	Макс. ширина подачи (единица, см. <_AMODE>)		
13	DZ	<_MID>	REAL	Макс. подача на глубину		
14	UXY	<_FAL>	REAL	Чистовой припуск плоскость		
15	UZ	<_FALD>	REAL	Чистовой припуск глубина		
16	F	<_FFP1>	REAL	Подача обработки		
17		<_CDIR>	INT	Направление фрезерования	0 =	синхронный ход
					1 =	противоход
18		<_VARI>	INT	Режим обработки		
				ПЕРВАЯ ПОЗИЦИЯ:	Обработка	
					1 =	черновая обработка
					2 =	чистовая обработка
					3 =	чистовая обработка края
					5 =	снятие фаски
				ВТОРАЯ ПОЗИЦИЯ:	Размер под ключ или длина граней	
					0 =	размер под ключ
					1 =	длина грани

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
19	FS	<_FS>	REAL	Ширина фаски (инкр.)
20	ZFS	<_ZFS>	REAL	Глубина врезания (острие инструмента) при снятии фаски (абс./инкр.) см. <_AMODE>
21		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: выбор обработки или только расчет стартовой точки
				1 = обычная обработка
22		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ: плоскость обработки G17/18/19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ: --- зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: --- зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: --- зарезервировано
				ПЯТАЯ ПОЗИЦИЯ: Технологическое масштабирование внутри масок цикла (Страница 870)
23		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ: конечная глубина (<_DP>)
				0 = абсолютная
				1 = инкрементальная
				ВТОРАЯ ПОЗИЦИЯ: единица для подачи в плоскости (<_MIDA>)
				0 = мм
				1 = % от диаметра инструмента
				ТРЕТЬЯ ПОЗИЦИЯ: глубина врезания при снятии фаски (<_ZFS>)
				0 = абсолютная
				1 = инкрементальная
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: закругление углов (<_RC>)
				0 = радиус
				1 = фаска

20.1.21 CYCLE81 - сверление, центрование

Синтаксис

```
CYCLE81(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<RFP>	REAL	Исходная точка (абс.)
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
4	Z1/Ø	<DP>	REAL	Глубина сверления (абс.) / диаметр центрования (абс.), см. <_GMODE>
5	Z1	<DPR>	REAL	Глубина сверления (инкр.)
6	DT	<DTB>	REAL	Время ожидания на конечной глубине сверления, см. <_AMODE>
7		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: центрование относительно глубины/диаметра
				0 = совместимость, глубина 1 = диаметр
8		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ: плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
9		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ: глубина сверления Z1 (абс./инкр.)
				0 = совместимость, из программирования DP/DPR
				1 = инкрементальная
				2 = абсолютная
				ВТОРАЯ ПОЗИЦИЯ: время ожидания на конечной глубине сверления DT в секундах/оборотах
				0 = совместимость, из знака DTB (> 0 секунд или < 0 оборотов)
				1 = в секундах 2 = в оборотах

20.1.22 CYCLE82 - сверление, цекование

Синтаксис

```
CYCLE82 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<RFP>	REAL	Исходная точка (абс.)
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
4	Z1	<DP>	REAL	Глубина сверления (абс.), см. <_AMODE>
5	Z1	<DPR>	REAL	Глубина сверления (инкр.), см. <_AMODE>
6	DT	<DTB>	REAL	Время ожидания на конечной глубине сверления, см. <_AMODE>
7		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: глубина сверления относительно острия/хвостовика
				0 = совместимость, острие
				1 = хвостовик
8		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ: плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: зарезервировано
				ПЯТАЯ ПОЗИЦИЯ: Технологическое масштабирование внутри масок цикла (Страница 870)
				0 = ввод: полный
				1 = ввод: простой

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
9		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ: глубина сверления Z1 (абс./инкр.)
				0 = совместимость из программирования DP/DPR
				1 = инкрементальная
				2 = абсолютная
				ВТОРАЯ ПОЗИЦИЯ: время ожидания DT на конечной глубине сверления в секундах/оборотах
				0 = совместимость, из знака DT (> 0 секунд/ < 0 оборотов)
				1 = в секундах
				2 = в оборотах
				ТРЕТЬЯ ПОЗИЦИЯ: глубина засверливания ZA абс./инкр.
				0 = инкрементальная
				1 = абсолютная
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: нормирование подачи засверливания
				0 = в % от подачи сверления
10		<_VARI>	INT	Режим обработки, засверливание/сквозное сверление
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: Сквозное сверление
				0 = сквозное сверление "Нет"
				1 = сквозное сверление "Да"
				ПЯТАЯ ПОЗИЦИЯ: Засверливание
				0 = засверливание "Нет"
				1 = засверливание "Да"
11	ZA	<S_ZA>	REAL	Глубина засверливания инкрементальная относительно исходной точки или абсолютная (см. <_AMODE> ТРЕТЬЯ ПОЗИЦИЯ)
12	FA	<S_FA>	REAL	Подача засверливания как значение или в % (в комбинации с <_AMODE>ЧЕТВЕРТАЯ ПОЗИЦИЯ)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
13	ZD	<S_ZD>	REAL	Остаточная глубина сверления инкрементальная относительно конечной глубины сверления или абсолютная (в комбинации с <_AMODE> ПЯТАЯ ПОЗИЦИЯ)
14	FD	<S_FD>	REAL	Остаточная подача сверления как значение или в % (в комбинации с <_AMODE> ШЕСТАЯ ПОЗИЦИЯ)

20.1.23 CYCLE83 - глубокое сверление

Синтаксис

CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>, <_DAM>, <DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>, <_DTD>, <_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<RFP>	REAL	Исходная точка (абс.)
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
4	Z1	<DP>	REAL	Конечная глубина сверления (абс.), см. <_AMODE>
5	Z1	<DPR>	REAL	Конечная глубина сверления (инкр.), см. <_AMODE>
6	D	<FDEP>	REAL	1. Глубина сверления (абс.), см. <_AMODE>
7	D	<FDPR>	REAL	1. Глубина сверления (инкр.), см. <_AMODE>
8	DF	<_DAM>	REAL	Величина депрессии / процент для каждой следующей подачи, см. <_AMODE>
9	DTB	<DTB>	REAL	Время ожидания на глубине сверления, см. <_AMODE>
10	DTS	<DTS>	REAL	Время ожидания в начальной точке (только при удалении стружки), см. <_AMODE>
11	FD1	<FRF>	REAL	Процент для подачи при первой подаче, см. <_AMODE>
12		<VARI>	INT	Режим обработки ПЕРВАЯ ПОЗИЦИЯ: ломка стружки/удаление стружки 0 = ломка стружки 1 = удаление стружки
13		<_AXN>	INT	Ось инструмента 0 = 3-я гео-ось 1 = 1-я гео-ось 2 = 2-я гео-ось > 2 3-я гео-ось
14	V1	<_MDEP>	REAL	Мин. подача на глубину (только при проценте для депрессии)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
15	V2	<_VRT>	REAL	Величина отвода после каждой обработки (только при ломке стружки)
				> 0 переменная величина отвода
				0 = стандартное значение 1 мм
16	DT	<_DTD>	REAL	Время ожидания на конечной глубине сверления, см. <_AMODE>
17	V3	<_DIS1>	REAL	Упреждающий зазор (только при удалении стружки, см. <_AMODE>
18		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: глубина сверления относительно острия/хвостовика
				0 = острие
				1 = хвостовик
19		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ: плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ: --- зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: --- зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: --- зарезервировано
				ПЯТАЯ ПОЗИЦИЯ: Технологическое масштабирование внутри масок цикла (Страница 870)
				0 = ввод: полный
				1 = ввод: простой

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
20		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				глубина сверления = конечная глубина сверления Z1 (абс./инкр.)
				0 = совместимость, из программирования <DP>/<DPR>
				1 = инкрементальная
				2 = абсолютная
				ВТОРАЯ ПОЗИЦИЯ:
				время ожидания на глубине сверления DTV в секундах/оборотах
				0 = совместимость из знака DTV (> 0 секунд или < 0 оборотов)
				1 = в секундах
				2 = в оборотах
				ТРЕТЬЯ ПОЗИЦИЯ:
				время ожидания в начальной точке DTS в секундах/оборотах
				0 = совместимость из знака DTS (> 0 секунд или < 0 оборотов)
				1 = в секундах
				2 = в оборотах
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				время ожидания на конечной глубине сверления DTD в секундах/оборотах
				0 = совместимость из знака DTD (> 0 секунд или < 0 оборотов)
				1 = в секундах
				2 = в оборотах
				ПЯТАЯ ПОЗИЦИЯ:
				1. Глубина сверления D (абс./инкр.)
				0 = совместимость, из программирования <FDEPF>/<DPR>
				1 = инкрементальная
				2 = абсолютная
				ШЕСТАЯ ПОЗИЦИЯ:
				Величина депрессии / процент <_DAM> для каждой следующей подачи
				0 = совместимость, из знака <_DAM>(> 0 величина депрессии или < 0 коэффициент от 0,001 до 1,0)
				1 = Величина депрессии
				2 = процент (от 0,001 до 100 %)
				СЕДЬМАЯ ПОЗИЦИЯ:
				упреждающий зазор V3 автоматический/ручной
				0 = совместимость, из знака <_DIS1> (= 0 автоматический или > 0 ручной)
				1 = автоматический (вычисляется в цикле)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
					2 = ручной (запрограммированное значение)
				ВОСЬМАЯ ПОЗИЦИЯ:	коэффициент подачи для первой подачи <FRF> как коэффициент/процент
					0 = совместимость, как коэффициент (от 0,001 до 1,0, FRF = 0 означает 100 %)
					1 = процент (от 0,001 до 999,999 %)

20.1.24 CYCLE84 - нарезание внутренней резьбы без компенсирующего патрона

Синтаксис

```
CYCLE84(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDAC>, <MPIT>,
<PIT>, <POSS>, <SST>, <SST1>, <_AXN>, <_PITA>, <_TECHNO>, <_VARI>,
<_DAM>, <_VRT>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
1	RP	<RTP>	REAL	Плоскость отвода (абс.)	
2	Z0	<RFP>	REAL	Исходная точка (абс.)	
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)	
4	Z1	<DP>	REAL	Глубина сверления = конечная глубина сверления (абс.), см. <_AMODE>	
5	Z1	<DPR>	REAL	Глубина сверления = конечная глубина сверления (инкр.), см. <_AMODE>	
6	DT	<DTB>	REAL	Время ожидания на глубине сверления в секундах	
7	SDE	<SDAC>	INT	Направление вращения после завершения цикла	
8		<MPIT>	REAL	Размер резьбы только для "ISO метрической" (шаг вычисляется при внутренней обработке)	
9	P	<PIT>	REAL	Шаг резьбы как значение, единицу см. <_PITA>	
10	$\alpha S^{(1)}$	<POSS>	REAL	Позиция шпинделя для ориентированного останова шпинделя	
11	S	<SST>	REAL	Скорость шпинделя для нарезания внутренней резьбы	
12	SR	<SST1>	REAL	Скорость шпинделя для отвода	
13		<_AXN>	INT	Ось сверления	0 = 3. гео-ось
					1 = 1-я гео-ось
					2 = 2-я гео-ось
					≥ 3 = 3-я гео-ось

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
14		<_PITA>	INT	Единица измерения шага резьбы (обработка <PIT> и <MPIT>)		
				0 =	шаг в мм	- обработка <MPIT>/<PIT>
				1 =	шаг в мм	- обработка <PIT>
				2 =	шаг в TPI	- обработка <PIT> (витков резьбы на дюйм)
				3 =	шаг в дюймах	- обработка <PIT>
				4 =	MODUL	- обработка <PIT>
15		<_TECHNO>	INT	Технология ¹⁾		
				ПЕРВАЯ ПОЗИЦИЯ:	параметры точного останова	
					0 =	активны параметры точного останова как перед вызовом цикла
					1 =	точный останов G601
					2 =	точный останов G602
					3 =	точный останов G603
				ВТОРАЯ ПОЗИЦИЯ:	предупреждение	
					0 =	активно с/без предупреждения как перед вызовом цикла
					1 =	с предупреждением FFWON
					2 =	без предупреждения FFWOF
				ТРЕТЬЯ ПОЗИЦИЯ:	ускорение	
					0 =	SOFT/BRISK/DRIVE активно как перед вызовом цикла
					1 =	с ограничением рывка SOFT
					2 =	без ограничения рывка BRISK
					3 =	уменьшенное ускорение DRIVE
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	MCALL шпиндельный режим	
					0 =	при MCALL снова активировать шпиндельный режим
					1 =	при MCALL оставаться в управлении по положению
16		<_VARI>	INT	Режим обработки		
				ПЕРВАЯ ПОЗИЦИЯ:	0 =	1 проход
					1 =	ломка стружки (нарезание глубокой внутренней резьбы)
					2 =	удаление стружки (нарезание глубокой внутренней резьбы)
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	ISO/SIEMENS режим для маски ввода не релевантно	
					0 =	вызов из ISO-совместимости
					1 =	вызов из контекста SIEMENS
17	D	<_DAM>	REAL	Макс. подача на глубину (только при удалении стружки/ломке стружки)		

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
18	V2	<_VRT>	REAL	Величина отвода после каждой обработки (только при ломке стружки), см. <_AMODE>
19		<_PTM>	STRING[15]	Строка как меркер для ввода шага резьбы ²⁾
20		<_PTAB>	STRING[5]	Строка для таблицы резьб ("" , "ISO", "BSW", "BSP", "UNC") ²⁾
21		<_PTABA>	STRING[20]	Строка для выбора в таблице резьб (к примеру, "M 10", "M 12", ...) ²⁾
22		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)
			ПЕРВАЯ ПОЗИЦИЯ:	зарезервировано
			ВТОРАЯ ПОЗИЦИЯ:	зарезервировано
23		<_DMODE>	INT	Режим индикации
			ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
			ВТОРАЯ ПОЗИЦИЯ:	зарезервировано
			ТРЕТЬЯ ПОЗИЦИЯ:	зарезервировано
			ЧЕТВЕРТАЯ ПОЗИЦИЯ:	режим совместимости (только для маски ввода для обратного перевода), если MD 52216 Бит0 = 1 ¹⁾
				0 = технологические параметры отображаются (совместимость): действуют TECHNO-параметры
				1 = технологические параметры не отображаются: действует технология "как запрограммировано перед вызовом цикла"
			ПЯТАЯ ПОЗИЦИЯ:	Технологическое масштабирование внутри масок цикла (Страница 870)
				0 = Ввод: полный
				1 = Ввод: простой

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
24		<_AMODE>	INT	Альтернативный режим		
				ПЕРВАЯ ПОЗИЦИЯ:	глубина сверления = конечная глубина сверления Z1 (абс./инкр.)	
					0 =	совместимость, из программирования <DP>/<DPR>
					1 =	инкрементальная
					2 =	абсолютная
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано	
				ТРЕТЬЯ ПОЗИЦИЯ:	зарезервировано	
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	направление вращения резьбы правое/левое	
					0 =	совместимость, из знака PIT/MPIT
					1 =	справа
					2 =	слева
				ПЯТАЯ ПОЗИЦИЯ:	зарезервировано	
				ШЕСТАЯ ПОЗИЦИЯ:	зарезервировано	
				СЕДЬМАЯ ПОЗИЦИЯ:	величина отвода после каждой обработки V2 вручную/автоматически	
0 =	совместимость, из программирования <_VRT> (> 0 переменное значение или ≤ 0 стандартное значение 1 мм/0,0394 дюйма)					
1 =	автоматически (стандартное значение 1 мм/0,0394 дюйма)					
2 =	вручную (как запрограммировано в V2)					

1) Поля технологии могут быть скрыты в зависимости от установочных данных
SD52216 \$MCS_FUNCTION_MASK_DRILL

2) Параметры 19, 20 и 21 используются только при выборе резьбы в таблицах резьб маски ввода. Доступ к таблицам резьб через определение цикла при выполнении цикла невозможен.

20.1.25 CYCLE85 - развертывание

Синтаксис

CYCLE85 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>, <_GMODE>, <_DMODE>, <_AMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<RFP>	REAL	Исходная точка (абс.)

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)		
4	Z1	<DP>	REAL	Глубина сверления (абс.), см. <_AMODE>		
5	Z1	<DPR>	REAL	Глубина сверления (инкр.), см. <_AMODE>		
6	DT	<DTB>	REAL	Время ожидания на конечной глубине сверления, см. <_AMODE>		
7	F	<FFR>	REAL	Подача		
8	FR	<RFF>	REAL	Подача при отводе		
9		<_GMODE>	INT	зарезервировано		
10		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)
11		<_AMODE>	INT	Альтернативный режим (сверление)		
				ПЕРВАЯ ПОЗИЦИЯ:	глубина сверления Z1 (абс./инкр.)	
					0 =	совместимость, из программирования DP/DPR
					1 =	инкрементальная
					2 =	абсолютная
				ВТОРАЯ ПОЗИЦИЯ:	время ожидания DT на конечной глубине сверления в секундах/оборотах	
					0 =	совместимость, из знака DT (> 0 секунд или < 0 оборотов)
					1 =	в секундах
2 =	в оборотах					

20.1.26 CYCLE86 - растачивание

Синтаксис

CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>, <RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<RFP>	REAL	Исходная точка (абс.)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)		
4	Z1	<DP>	REAL	Глубина сверления (абс.), см. <_AMODE>		
5	Z1	<DPR>	REAL	Глубина сверления (инкр.), см. <_AMODE>		
6	DT	<DTB>	REAL	Время ожидания на конечной глубине сверления, см. <_AMODE>		
7	DIR	<SDIR>	INT	Направление вращения шпинделя	3 =	M3
					4 =	M4
8	Dx	<RPA>	REAL	Величина отвода в направлении X		
9	DY	<RPO>	REAL	Величина отвода в направлении Y		
10	DZ	<RPAP>	REAL	Величина отвода в направлении Z		
11	SPOS	<POSS>	REAL	Позиция шпинделя для отвода (для ориентированного останова шпинделя, в градусах)		
12		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)		
				ПЕРВАЯ ПОЗИЦИЯ:	режим подъема	
					0 =	подъем, совместимость
					1 =	Нет подъема
13		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)
14		<_AMODE>	INT	Альтернативный режим		
				ПЕРВАЯ ПОЗИЦИЯ:	глубина сверления Z1 (абс./инкр.)	
					0 =	совместимость, из программирования <DP>/<DPR>
					1 =	инкрементальная
					2 =	абсолютная
				ВТОРАЯ ПОЗИЦИЯ:	время ожидания на конечной глубине сверления DT в секундах/оборотах	
					0 =	совместимость, из знака DT (> 0 секунд или < 0 оборотов)
					1 =	в секундах
					2 =	в оборотах

20.1.27 CYCLE92 - отрез

Синтаксис

```
CYCLE92(<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>, <_SV1>,
<_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>, <_VARI>, <_DN>,
<_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
1	X0	<_SPD>	REAL	Исходная точка (абс., всегда диаметр)		
2	Y0	<_SPL>	REAL	Исходная точка (абс.)		
3	X1	<_DIAG1>	REAL	Глубина для снижения скорости, см. <_AMODE>(ПЕРВАЯ ПОЗИЦИЯ)		
4	X2	<_DIAG2>	REAL	Конечная глубина, см. <_AMODE>(ВТОРАЯ ПОЗИЦИЯ)		
5	R/FS	<_RC>	REAL	Радиус закругления или ширина фаски, см. <_AMODE>(ЧЕТВЕРТАЯ ПОЗИЦИЯ)		
6	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)		
7	S	<_SV1>	REAL	Постоянная скорость шпинделя, см. <_AMODE> (ПЯТАЯ ПОЗИЦИЯ)		
	V			Постоянная скорость резания		
8	SV	<_SV2>	REAL	Макс. скорость при постоянной скорости резания		
9	DIR	<_SDAC>	INT	Направление вращения шпинделя	3 =	для M3
					4 =	для M4
10	F	<_FF1>	REAL	Подача до глубины для снижения скорости		
11	FR	<_FF2>	REAL	Уменьшенная подача до конечной глубины		
12	SR	<_SS2>	REAL	Уменьшенная скорость до конечной глубины		
13	XM	<_DIAGM>	REAL	Глубина выдвижения приемника деталей (абс., всегда диаметр)		
14		<_VARI>	INT	Режим обработки		
				ПЕРВАЯ ПОЗИЦИЯ:	отвод	
					0 =	отвод на <_SPD> + <_SDIS>
					1 =	без отвода в конце
				ВТОРАЯ ПОЗИЦИЯ:	приемник деталей	
					0 =	нет, не выполнять M-команду
					1 =	да, вызов CUST_TCHCYC(101)- выдвинуть деталеприемник, CUST_TCHCYC(102)- закрыть деталеприемник
15		<_DN>	INT	D-номер для 2-го резца, если не запрограммирован ⇒ D+1		

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
20		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
21		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				глубина для снижения скорости (<_DIAG1>)
				0 = абсолютная, значение поперечной оси в диаметре
				1 = инкрементальная, значение поперечной оси в радиусе
				ВТОРАЯ ПОЗИЦИЯ:
				конечная глубина (<_DIAG2>)
				0 = абсолютная, значение поперечной оси в диаметре
				1 = инкрементальная, значение поперечной оси в радиусе
				ТРЕТЬЯ ПОЗИЦИЯ:
				зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				радиус/фаска (<_RC>)
				0 = Радиус
				1 = Фаска
				ПЯТАЯ ПОЗИЦИЯ:
				скорость шпинделя/скорость резания (<_SV1>)
				0 = постоянная скорость шпинделя
				1 = постоянная скорость резания

20.1.28 CYCLE95 - съём припуска вдоль контура

Синтаксис

CYCLE95 (<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>, <_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	CON	<NPP>	STRING [140]	Имя контура
2	D	<MID>	REAL	Макс. подача на глубину при черновой обработке, см. <_GMODE>

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
3	UZ	<FALZ>	REAL	Чистовой припуск в Z	
4	UX	<FALX>	REAL	Чистовой припуск в X	
5	U	<FAL>	REAL	Чистовой припуск параллельно контуру (действует в обеих осях)	
6	F	<FF1>	REAL	Подача для черновой обработки	
7	FY	<FF2>	REAL	Подача врезания поднутрения	
8	FS	<FF3>	REAL	Чистовая подача	
9		<_VARI>	INT	Режим обработки	
				ПЕРВАЯ и ВТОРАЯ позиция:	
					1 = черновая обработка, вдоль, снаружи
					2 = черновая обработка, поперек, снаружи
					3 = черновая обработка, вдоль, внутри
					4 = черновая обработка, поперек, внутри
					5 = чистовая обработка, вдоль, снаружи
					6 = чистовая обработка, поперек, снаружи
					7 = чистовая обработка, вдоль, внутри
					8 = чистовая обработка, поперек, внутри
					9 = комплексная обработка, вдоль, снаружи
					10 = комплексная обработка, поперек, снаружи
					11 = комплексная обработка, вдоль, внутри
					12 = комплексная обработка, поперек, внутри
				ТРЕТЬЯ ПОЗИЦИЯ:	
					0 = с возвратом по контуру, без остаточных углов
					1 = с прямым возвратом
					2 = возврат только до прежней точки резания, возможны остаточные углы
10	DT	<DT>	REAL	Время ожидания при прерывании подачи	
11	DI	<DAM>	REAL	Интервал прерываний подачи	

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
12	VRT	<_VRT>	REAL	Путь отвода от контура		
					0 =	внутри системы используется путь отвода в 1 мм независимо от дюймовой/метрической системы единиц
					> 0 =	путь отвода
13		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)		
				ПЕРВАЯ ПОЗИЦИЯ:	Нормирование глубины подачи	
					0 =	глубина подачи рассчитывается согласно G-группе DIAMON/DIAMOF
					1 =	глубина подачи действует как значение радиуса (независимо от DIAMON/DIAMOF)
14		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:		
					0 =	режим совместимости: имя контура стоит в NPP
					1 =	имя контура программируется в CYCLE62 и передается в _SC_CONT_NAME

20.1.29 CYCLE98 - цепочка резьб

Синтаксис

```
CYCLE98 (<_PO1>, <_DM1>, <_PO2>, <_DM2>, <_PO3>, <_DM3>, <_PO4>,
<_DM4>, <APP>, <ROP>, <TDEP>, <FAL>, <_IANG>, <NSP>, <NRC>, <NID>,
<_PP1>, <_PP2>, <_PP3>, <_VARI>, <_NUMTH>, <_VRT>, <_MID>, <_GDEP>,
<_IFLANK>, <_PITA>, <_PITM1>, <_PITM2>, <_PITM3>, <_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
1	Z0	<_PO1>	REAL	Исходная точка в Z (абс.)		
2	X0	<_DM1>	REAL	Исходная точка в X (абс.), в диаметре		
3	Z1	<_PO2>	REAL	Промежуточная точка 1 в Z (абс./инкр.), см. <_AMODE>(ПЕРВАЯ ПОЗИЦИЯ)		
4	X1	<_DM2>	REAL	Промежуточная точка 1 в X (абс./инкр.), см. <_AMODE>(ВТОРАЯ ПОЗИЦИЯ) или		
	X1α			Наклон резьбы 1 (-90° до 90°) абс. всегда диаметр, инкр. всегда радиус		
5	Z2	<_PO3>	REAL	Промежуточная точка 2 в Z (абс./инкр.), см. <_AMODE>(ТРЕТЬЯ ПОЗИЦИЯ)		
6	X2	<_DM3>	REAL	Промежуточная точка 2 в X (абс./инкр.), см. <_AMODE>(ЧЕТВЕРТАЯ ПОЗИЦИЯ) или		
	X2α			Наклон резьбы 2 (-90° до 90°) абс. всегда диаметр, инкр. всегда радиус		
7	Z3	<_PO4>	REAL	Конечная точка в Z (абс./инкр.), см. <_AMODE> (ПЯТАЯ ПОЗИЦИЯ)		
8	X3	<_DM4>	REAL	Конечная точка в X, (абс./инкр.), см. <_AMODE>(ШЕСТАЯ ПОЗИЦИЯ) или		
	X3α			Наклон резьбы 3 (-90° до 90°) абс. всегда диаметр, инкр. всегда радиус		
9	LW	<APP>	REAL	Заход резьбы (инкр., ввод без знака)		
10	LR	<ROP>	REAL	Выход резьбы (инкр., ввод без знака)		
11	H1	<TDEP>	REAL	Глубина резьбы (инкр., ввод без знака)		
12	U	<FAL>	REAL	Чистовой припуск в X и Z		
13	DP	<_IANG>	REAL	Наклон подачи как интервал или угол, см. <_AMODE> (СЕДЬМАЯ ПОЗИЦИЯ)		
	αP			Наклон подачи действует согласно установке Параметр <_VARI>(ТРЕТЬЯ ПОЗИЦИЯ).		
				Определение для <_VARI_ТРЕТЬЯ ПОЗИЦИЯ = 0 - режим совместимости:		
				> 0 =	боковая подача по одной стороне	
				0 =	подача вертикально в резьбе	
				< 0 =	боковая подача по разным сторонам	
				Определение для <_VARI_ТРЕТЬЯ ПОЗИЦИЯ>0:		
				> 0 =	подача по положительной боковой стороне	
0 =			подача по центру			
			< 0 =	подача по отрицательной боковой стороне		
14	α0	<NSP>	REAL	Смещение начального угла для 1-го витка резьбы		
15		<NRC>	INT	Число черновых проходов, см. <_VARI>(ПЯТАЯ ПОЗИЦИЯ)		

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
16	NN	<NID>	INT	Число холостых проходов	
17	P0	<_PP1>	REAL	Шаг резьбы на 1-м участке резьбы, см. <_PITA>	
18	P1	<_PP2>	REAL	Шаг резьбы на 2-м участке резьбы, см. <_PITA>	
19	P2	<_PP3>	REAL	Шаг резьбы на 3-м участке резьбы, см. <_PITA>	
20		<_VARI>	INT	Обработка	
				ПЕРВАЯ ПОЗИЦИЯ:	технология
					1 = наружная резьба с линейной подачей
					2 = внутренняя резьба с линейной подачей
					3 = наружная резьба с дегрессивной подачей, поперечное сечение резания остается постоянным
					4 = наружная резьба с дегрессивной подачей, поперечное сечение резания остается постоянным
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:	тип подачи
					0 = режим совместимости для <_IANG>
					1 = подача с одной стороны
					2 = попеременная подача
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	зарезервировано
				ПЯТАЯ ПОЗИЦИЯ:	альтернативная подача на глубину
					0 = совместимость, задание числа черновых проходов (<_NRC>)
					1 = задание значения для 1-й подачи (<_MID>)
				ШЕСТАЯ ПОЗИЦИЯ:	режим обработки
					0 = совместимость (черновая и чистовая обработка)
					1 = черновая обработка
					2 = чистовая обработка
					3 = черновая и чистовая обработка
				СЕДЬМАЯ ПОЗИЦИЯ:	последовательность обработки для многозаходной резьбы
					0 = восходящая последовательность витков
					1 = нисходящая последовательность витков
21	n	<_NUMTH>	INT	Число витков резьбы	

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
22		<_VRT>	REAL	Интервал обратного хода (инкр.)		
					0 =	внутри системы используется путь отвода в 1 мм независимо от дюймовой/метрической системы единиц
					> 0 =	путь отвода
23	D1	<_MID>	REAL	Первая подача, см. <_VARI>(ПЯТАЯ ПОЗИЦИЯ)		
24	DA	<_GDEP>	REAL	Глубина смены захода (действует только при "многозаходная")		
					0 =	не учитывать глубину смены захода
					> 0 =	учитывать глубину смены захода
25		<_IFLANK>	REAL	Наклон подачи как ширина (только для интерфейса)		
26		<_PITA>	INT	Нормирование шага резьбы		
					0 =	режим совместимости для шага резьбы, обработка <_PP1> до <_PP3> как прежде согласно активной метрической/дюймовой системе
					1 =	шаг в мм
					2 =	шаг в TPI (витков резьбы на дюйм)
					3 =	шаг в дюймах
					4 =	MODUL
27		<_PITM1>	STRING[15]	Строка как меркер для ввода шага резьбы (только для интерфейса)		
28		<_PITM2>	STRING[15]	Строка как меркер для ввода шага резьбы (только для интерфейса)		
29		<_PITM3>	STRING[15]	Строка как меркер для ввода шага резьбы (только для интерфейса)		
30		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:	---	зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:	---	зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	---	зарезервировано
				ПЯТАЯ ПОЗИЦИЯ:	Технологическое масштабирование внутри масок цикла (Страница 870)	
					0 =	Ввод: полный
					1 =	Ввод: простой

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
31		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				1. промежуточная точка в Z (Z1)
				0 = абсолютная
				1 = инкрементальная
				ВТОРАЯ ПОЗИЦИЯ:
				1. промежуточная точка в X (X1)
				0 = абсолютная
				1 = инкрементальная
				2 = α
				ТРЕТЬЯ ПОЗИЦИЯ:
				2. промежуточная точка в Z (Z2)
				0 = абсолютная
				1 = инкрементальная
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				2. промежуточная точка в X (X2)
				0 = абсолютная
				1 = инкрементальная
				2 = α
				ПЯТАЯ ПОЗИЦИЯ:
				конечная точка в Z (Z3)
				0 = абсолютная
				1 = инкрементальная
				ШЕСТАЯ ПОЗИЦИЯ:
				конечная точка в X (X3)
				0 = абсолютная
				1 = инкрементальная
				2 = α
				СЕДЬМАЯ ПОЗИЦИЯ:
				выбор наклона подачи как угла или ширины
				0 = угол подачи <_IANG>
				1 = наклон подачи <_IFLANK>
				ВОСЬМАЯ ПОЗИЦИЯ:
				однозаходная/многозаходная
				0 = режим совместимости (начальный угол <_NSP>обрабатывается)
				1 = однозаходная (со смещение начального угла <_NSP>)
				2 = многозаходная

20.1.30 CYCLE99 - нарезание резьбы резцом

Синтаксис

```
CYCLE99(<_SPL>, <_SPD>, <_FPL>, <_FPD>, <_APP>, <_ROP>, <_TDEP>,
<_FAL>, <_IANG>, <_NSP>, <_NRC>, <_NID>, <_PIT>, <_VARI>, <_NUMTH>,
<_SDIS>, <_MID>, <_GDEP>, <_PIT1>, <_FDEP>, <_GST>, <_GUD>,
```

20.1 Технологические циклы

<_IFLANK>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>, <_S_XRS>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
1	Z0	<_ SPL>	REAL	Исходная точка (абс.)		
2	X0	<_ SPD>	REAL	Исходная точка (абс., всегда диаметр)		
3	Z1	<_ FPL>	REAL	Конечная точка в комбинации с <_ AMODE>(ПЕРВАЯ ПОЗИЦИЯ)		
4	X1	<_ FPD>	REAL	Конечная точка, в комбинации с <_ AMODE>(ВТОРАЯ ПОЗИЦИЯ)		
5	LW/LW2	<_ APP>	REAL	Заход резьбы в комбинации с <_ AMODE> (ТРЕТЬЯ ПОЗИЦИЯ) или Вход резьбы = выход резьбы в комбинации с <_ AMODE>(ТРЕТЬЯ ПОЗИЦИЯ)		
6	LR	<_ ROP>	REAL	Выход резьбы		
7	H1	<_ TDEP>	REAL	Глубина резьбы		
8	U	<_ FAL>	REAL	Чистовой припуск в X и Z		
9	DP	<_ IANG>	REAL	Наклон подачи как интервал или угол, в комбинации с <_ AMODE>(ЧЕТВЕРТАЯ ПОЗИЦИЯ)		
	αP			> 0 =	подача по положительной боковой стороне	
				< 0 =	подача по отрицательной боковой стороне	
				0 =	подача по центру	
10	α0	<_ NSP>	REAL	Смещение начального угла (действует только при "однозаходная")		
11	ND	<_ NRC>	INT	Число черновых проходов, в комбинации с <_ VARI>(ПЯТАЯ ПОЗИЦИЯ)		
12	NN	<_ NID>	INT	Число холостых проходов		
13	P	<_ PIT>	REAL	Шаг резьбы как значение, в комбинации с <_ PITA>		

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
14		<_VARI>	INT	Режим обработки	
				ПЕРВАЯ ПОЗИЦИЯ:	технология
					1 = наружная резьба с линейной подачей
					2 = внутренняя резьба с линейной подачей
					3 = наружная резьба с депрессивной подачей, поперечное сечение резания остается постоянным
					4 = наружная резьба с депрессивной подачей, поперечное сечение резания остается постоянным
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:	тип подачи
					1 = подача с одной стороны
					2 = попеременная подача
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	зарезервировано
				ПЯТАЯ ПОЗИЦИЯ:	альтернативная подача на глубину
					0 = задание числа черновых проходов (<_NRC>)
					1 = задание значения для 1-й подачи (<_MID>)
				ШЕСТАЯ ПОЗИЦИЯ:	режим обработки
					1 = Черновая обработка
					2 = Чистовая обработка
					3 = Черновая и чистовая обработка
				СЕДЬМАЯ ПОЗИЦИЯ:	последовательность обработки для многозаходной резьбы
					0 = восходящая последовательность витков
					1 = нисходящая последовательность витков
15	n	<_NUMTH>	INT	Число витков резьбы	
16	VR	<_SDIS>	REAL	Интервал обратного хода, инкр.	
17	D1	<_MID>	REAL	Первая глубина подачи, в комбинации с <_VARI>(ПЯТАЯ ПОЗИЦИЯ)	
18	DA	<_GDEP>	REAL	Глубина смены захода (действует только при "многозаходная")	
				0 =	не учитывать глубину смены захода
				> 0 =	учитывать глубину смены захода

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
19	g	<_PIT1>	REAL	Изменение шага на оборот
				0 = постоянный шаг резьбы (G33)
				> 0 = увеличение шага резьбы (G34)
				< 0 = уменьшение шага резьбы (G35)
20		<_FDEP>	REAL	Глубина врезания (вводится без знака)
21	N1	<_GST>	INT	Стартовый заход N1 = 1...N, в комбинации с <_AMODE>(ШЕСТАЯ ПОЗИЦИЯ)
22		<_GUD>	INT	зарезервировано
23		<_IFLANK>	REAL	Наклон подачи как ширина (только для интерфейса)
24		<_PITA>	INT	Единица измерения шага резьбы (обработка PIT и/или MPIT)
				0 = шаг в мм - обработка MPIT/PIT
				1 = шаг в мм - обработка PIT
				2 = шаг в TPI - обработка PIT (витков резьбы на дюйм)
				3 = шаг в дюймах - обработка PIT
				4 = MODUL - обработка PIT
25		<_PITM>	STRING[15]	Строка как меркер для ввода шага резьбы (только для интерфейса) ¹⁾
26		<_PTAB>	STRING[20]	Строка для таблицы резьб (только для интерфейса) ¹⁾
27		<_PTABA>	STRING[20]	Строка для выбора таблицы резьб (только для интерфейса) ¹⁾
28		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ: плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ: тип резьбы
				0 = продольная резьба
				1 = спиральная резьба
				2 = коническая резьба
				ТРЕТЬЯ ПОЗИЦИЯ: --- зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: --- зарезервировано
				ПЯТАЯ ПОЗИЦИЯ: Технологическое масштабирование внутри масок цикла (Страница 870)
				0 = Ввод: полный
				1 = Ввод: простой

№	Параметр в маске	Параметр внутренний	Тип данных	Значение			
29		<_AMODE>	INT	Альтернативный режим			
				ПЕРВАЯ ПОЗИЦИЯ:	длина резьбы в Z		
					0 =	абсолютная	
					1 =	инкрементальная	
				ВТОРАЯ ПОЗИЦИЯ:	длина резьбы в X		
					0 =	абсолютная, значение поперечной оси в диаметре	
					1 =	инкрементальная, значение поперечной оси в радиусе	
					2 =	α	
				ТРЕТЬЯ ПОЗИЦИЯ:	нормирование захода/входного участка <_APP>		
					0 =	заход резьбы <_APP>	
					1 =	вход резьбы = выход резьбы <_APP>= -<_ROP>	
					2 =	задание входа резьбы <_APP> = -<_APP>	
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	выбор наклона подачи как угла или ширины		
					0 =	угол подачи <_IANG>	
					1 =	наклон подачи <_IFLANK>	
				ПЯТАЯ ПОЗИЦИЯ:	однозаходная/многозаходная		
					0 =	однозаходная (со смещение начального угла <_NSP>)	
					1 =	многозаходная	
				ШЕСТАЯ ПОЗИЦИЯ:	стартовый заход <_GST>		
					0 =	полная обработка	
1 =	запустить обработку с этого витка						
2 =	обработать только этот виток						
СЕДЬМАЯ ПОЗИЦИЯ:	компенсация провисания при продольной резьбе						
	0 =	высота сегмента сферической резьбы XS					
	1 =	радиус сферической резьбы RS					
30	XS / RS	<_S_XRS>	REAL	Компенсация провисания при продольной резьбе в комбинации с <_AMODE>: СЕДЬМАЯ ПОЗИЦИЯ			

Примечание

¹⁾ Параметры <_PITM>, <_PTAB> и <_PTABA> используются только при выборе резьбы в таблицах резьб маски ввода.

Доступ к таблицам резьб через определение цикла при выполнении цикла невозможен.

20.1.31 CYCLE435 - установка системы координат правящего инструмента

Синтаксис

```
CYCLE435(<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>,
<S_PVL>, <S_PD>, <S_PL>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1		<_T>	STRING[32]	Имя инструмента шлифовального круга
2		<_DD>	INT	Номер резца шлифовального круга
3		<S_TA>	STRING[32]	Исходная точка правящего инструмента - наименование правящего инструмента
4		<S_DA>	INT	Номер резца правящего инструмента
5		<S_AD>	REAL	Величина правки, диаметр
6		<S_AL>	REAL	Величина правки, плоскость
7		<S_PVD>	REAL	Смещение при профилировании, диаметр
8		<S_PVL>	REAL	Смещение при профилировании, плоскость
9		<S_PD>	REAL	Припуск при профилировании, диаметр
10		<S_PL>	REAL	Припуск при профилировании, плоскость
11		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				активный инструмент в конце цикла
				0 = правящий инструмент активен
				1 = шайба активна

20.1.32 CYCLE495 - Профилирование

Синтаксис

```
CYCLE495(<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>,
<S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1		<_T>	STRING[20]	Имя инструмента шлифовального круга
2		<_DD>	INT	Номер резца шлифовального круга
3		<_SC>	REAL	Величина отвода для обхода препятствий, инкрементальная
4		<_F>	REAL	Подача при профилировании

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
5		<_VARI>	INT	Режим обработки	
				ПЕРВАЯ ПОЗИЦИЯ:	тип профилирования
					1 = параллельно осям
					2 = параллельно контуру
				ВТОРАЯ ПОЗИЦИЯ:	Направление обработки
					0 = протягивающий возможно при положении резцов 1 - 4
					1 = толкающий возможно при положении резцов 1 - 4
					2 = чередующийся возможно при положении резцов 1 - 8
					3 = начало → конец возможно при положении резцов 1 - 8
					4 = конец → начало возможно при положении резцов 1 - 8
				ТРЕТЬЯ ПОЗИЦИЯ:	направление подачи
					1 = подача X- при G18 или Y- при G19
					2 = подача X+ при G18 или Y+ при G19
					3 = подача Z- при G18 и при G19
					4 = подача Z+ при G18 и при G19
6		<_D>	REAL	Величина правки при профилировании параллельно оси	
7		<_DX>	REAL	Величина правки X при G18 или Y при G19 при профилировании параллельно контуру	
8		<_DZ>	REAL	Величина правки Z при G18 и G19 при профилировании параллельно контуру	
9		<S_PA>	REAL	Припуск на профилирование	
10		<S_N>	INT	Число ходов в программе профилирования	
11		<_DMODE>	INT	Режим индикации	
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19
					0 = совместимость, остается активной действующая перед вызовом цикла плоскость
					1 = G17 (активна только в цикле)
					2 = G18 (активна только в цикле)
					3 = G19 (активна только в цикле)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
12		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				выбор профилирования Заново/Продолжить
				1 = заново
				2 = продолжить
				ВТОРАЯ ПОЗИЦИЯ:
				выбор припуска на профилирование
				0 = от контура заготовки до самой глубокой точки контура
				1 = от контура заготовки до самой высокой точки контура
13		<S_FW>	REAL	Задний угол правящего инструмента
14		<S_HW>	REAL	Угол зажима правящего инструмента

20.1.33 CYCLE800 - поворот

Синтаксис

CYCLE800(<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>, <_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
1		<_FR>	INT	Режим отвода:
				0 = нет отвода
				1 = отвод оси станка Z
				2 = отвод оси станка Z и затем XY
				3 = зарезервировано
				4 = отвод в направлении инструмента макс.
				5 = отвод в направлении инструмента инкр.
2		<_TC>	STRING[32]	Имя блока данных поворота:
				"" (нет имени), если имеется только 1 блок данных поворота
				"0" отмена блока данных поворота (удаление фрейма поворота)

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
3		<_ST>	INT	Состояние трансформаций
				ПЕРВАЯ ПОЗИЦИЯ:
				0 = новая, повернутая плоскость удаляется и заново рассчитывается с текущими параметрами
				1 = аддитивная, повернутая плоскость добавляется аддитивно к активной повернутой плоскости
				ВТОРАЯ ПОЗИЦИЯ:
				слежение за острием инструмента да/нет (активно, то если функция установлена при вводе в эксплуатацию поворота)
				0 = не следить за острием инструмента
				1 = следить за острием инструмента (TRAORI)
				ТРЕТЬЯ ПОЗИЦИЯ:
				установить / точно установить инструмент (функция отображается в маске ввода ПОВОРОТ инструмента)
				0 = не устанавливать инструмент
				1 = установить инструмент (преимущественно радиусные фрезы)
				2 = точно установить токарный инструмент (если кинематика оси В для токарной технологии установлена при вводе поворота в эксплуатацию)
				3 = точно установить фрезерный инструмент (если кинематика оси В для токарной технологии установлена при вводе поворота в эксплуатацию)
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				внутренний параметр Поворот в JOG
				ПЯТАЯ ПОЗИЦИЯ:
				см. параметр Направление <_DIR>
				0 = поворот "да"
				1 = поворот "нет", направление "минус" ³⁾
				2 = поворот "нет", направление "плюс" ³⁾
				ШЕСТАЯ ПОЗИЦИЯ:
				см. параметр Направление <_DIR>
				0 = Совместимость
				1 = Выбор направления "минус" оптимизирован (только для интерфейса пользователя) ⁴⁾

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение		
					2 =	Выбор направления "плюс" оптимизирован (только для интерфейса пользователя) ⁴⁾
4		<_MODE> ⁵⁾	INT	Режим поворота: Обработка угла поворота и последовательности поворота (битовая кодировка!)		
				Бит: 7 6	0 0:	поворот каждой осью -> см. параметры <_A>, <_B>, <_C>
					0 1:	пространственный угол -> см. параметры <_A>, <_B> ¹⁾
					1 0:	угол проецирования -> см. параметры <_A>, <_B>, <_C> ¹⁾
					1 1:	режим поворота Круговые оси напрямую -> см. параметры <_A>, <_B> ¹⁾
				Бит: 5 4 3 2 1 0 (для пространственных углов без значения!)	x x x x 0 1	1-й поворот _A вокруг X
					x x x x 1 0	1-й поворот _A вокруг Y
					x x x x 1 1	1-й поворот _A вокруг Z
					x x 0 1 x x	2-й поворот _B вокруг X
					x x 1 0 x x	2-й поворот _B вокруг Y
					x x 1 1 x x	2-й поворот _B вокруг Z
					0 1 x x x x	3-й поворот _C вокруг X
					1 0 x x x x	3-й поворот _C вокруг Y
					1 1 x x x x	3-й поворот _C вокруг Z
5	X0	<_X0>	REAL	Исходная точка X перед вращением		
6	Y0	<_Y0>	REAL	Исходная точка Y перед вращением		
7	Z0	<_Z0>	REAL	Исходная точка Z перед вращением		
8	X(A)	<_A>	REAL	1-е вращение согласно настройкам параметров <_MODE>		
9	Y(B)	<_B>	REAL	2-е вращение согласно настройкам параметров <_MODE>		
10	Z(C)	<_C>	REAL	3-е вращение согласно настройкам параметров <_MODE>		
11	X1	<_X1>	REAL	Исходная точка X после вращения		
12	Y1	<_Y1>	REAL	Исходная точка Y после вращения		
13	Z1	<_Z1>	REAL	Исходная точка Z после вращения		
14	- или +	<_DIR>	INT	Запустить движение перемещения круговых осей (по умолчанию = -1!)		
					-1 =	позиционировать на меньшее значение круговой оси 1 или 2 ²⁾
					+1 =	позиционировать на большее значение круговой оси 1 или 2 ²⁾
					0 =	нет поворота (только расчет фрейма поворота) ^{1) 3)}
15	FR	<_FR_I>	REAL	Значение (инкр.) отвод в направлении инструмента инкрементальный		

№	Параметр в маске	Параметр внутренних	Тип данных	Объяснение
16		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:
				представление бета-значения при точной установке инструмента
				0 = Значение
				1 = Указатель

Примечание

Если следующие передаваемые параметры запрограммированы косвенно (как параметры), то обратный перевод маски ввода не выполняется: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

¹⁾ Выбор возможен, если функция установлена при ВВОДЕ В ЭКСПЛУАТАЦИЮ ПОВОРОТА.

²⁾ Выбор возможен, если при ВВОДЕ В ЭКСПЛУАТАЦИЮ ПОВОРОТА зависимость направления перемещения установлена на круговую ось 1 или 2.

Нет поля выбора при Зависимость направления перемещения нет

³⁾ Выбор поворота "нет" может быть скрыт SD 55221 Бит 0

Поворот "нет" направление "минус" соответствует <_DIR>= 0 и _ST ПЯТАЯ ПОЗИЦИЯ = 1

Поворот "нет" направление "плюс" соответствует <_DIR>= 0 и _ST ПЯТАЯ ПОЗИЦИЯ = 2

⁴⁾ Выбор направления круговой оси 1 или 2 осуществляется и тогда, когда круговая ось с зависимостью направления перемещения находится в позиции полюса (значение позиции равно нулю).

⁵⁾ Пример кодировки: вращение каждой осью, последовательность вращения ZYX

Двоичная: 00011011; десятичная: 27

Идентификаторы осей XYZ соответствуют гео-осям канала ЧПУ. Повороты вокруг осей XYZ могут выполняться по отдельности. Например, последовательность вращения вокруг ZXZ в одном вызове CYCLE800 не разрешена

20.1.34 CYCLE801 - Решетка или рамка

Синтаксис

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

Параметр

№	Параметр в маске	Параметр внутр.	Тип данных	Значение		
1	X0	<_SPCA>	REAL	Исходная точка для образца позиций (решетка/рамка) в 1-й оси (абс.)		
2	Y0	<_SPCO>	REAL	Исходная точка для образца позиций (решетка/рамка) во 2-й оси (абс.)		
3	α0	<_STA>	REAL	Базовый угол поворота (угол к 1-й оси)	< 0 =	Вращение по часовой стрелке
					> 0 =	Вращение против часовой стрелки
4	L1	<_DIS1>	REAL	Расстояние между колонками (интервал позиции 1-й оси, вводится без знака)		
5	L2	<_DIS2>	REAL	Расстояние между рядами (интервал позиции 2-й оси, вводится без знака)		
6	N1	<_NUM1>	INT	Число колонок		
7	N2	<_NUM2>	INT	Число рядов		
8		<_VARI>	INT	Режим обработки		
				ПЕРВАЯ ПОЗИЦИЯ:	образец позиции	
					0 =	Решетка
					1 =	Рамка
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано	
				ТРЕТЬЯ ПОЗИЦИЯ:	зарезервировано	
9		<_UMODE>	INT	зарезервировано		
10	αX	<_ANG1>	REAL	Угол среза к 1-й оси (наклонное положение рядов относительно 1-й оси)		
					< 0 =	измерение по часовой стрелке (от 0 до -90 градусов)
					> 0 =	измерение против часовой стрелки (от 0 до 90 градусов)
11	αY	<_ANG2>	REAL	Угол среза ко 2-й оси (наклонное положение рядов относительно 2-й оси)		
					< 0 =	измерение по часовой стрелке (от 0 до -90 градусов)
					> 0 =	измерение против часовой стрелки (от 0 до 90 градусов)

№	Параметр в маске	Параметр внутр.	Тип данных	Значение
12		<_HIDE>	STRING [200]	Пропущенные позиции - макс. 198 знаков - Указание текущего номера позиции, к примеру, "1,3" (позиции 1 и 3 не выполняются)
13		<_NSP>	INT	зарезервировано
14		<_DMODE>	INT	Режим индикации
		ПЕРВАЯ ПОЗИЦИЯ:		плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)

20.1.35 CYCLE802 - любые позиции

Синтаксис

```
CYCLE802 (<_XA>, <_YA>, <_X0>, <_Y0>, <_X1>, <_Y1>, <_X2>, <_Y2>,
<_X3>, <_Y3>, <_X4>, <_Y4>, <_X5>, <_Y5>, <_X6>, <_Y6>, <_X7>, <_Y7>,
<_X8>, <_Y8>, <_VARI>, <_UMODE>, <_DMODE>, <S_ABA>, <S_AB0>,
<S_AB1>, <S_AB2>, <S_AB3>, <S_AB4>, <S_AB5>, <S_AB6>, <S_AB7>,
<S_AB8>)
```

Параметр

№	Параметр в маске	Параметр внутр.	Тип данных	Значение
1		<_XA>	INT	Альтернатива для всех X-позиций (9-значное десятичное значение) Число разрядов: 876543210 (разряд соответствует позиции сверления Xn)
		Значение разряда:		1 = абсолютное (1-я запрограммированная позиция всегда абсолютная!)
				2 = инкрементальное

№	Параметр в маске	Параметр внутр.	Тип данных	Значение
2		<_YA>	INT	Альтернатива для всех Y-позиций (9-значное десятичное значение) Число разрядов: 876543210 (разряд соответствует позиции сверления Y _n)
				Значение разряда:
				1 = абсолютное (1-я запрограммированная позиция всегда абсолютная!) 2 = инкрементальное
3	X0	<_X0>	REAL	1. позиция X
4	Y0	<_Y0>	REAL	1. позиция Y
5	X1	<_X1>	REAL	2. позиция X
6	Y1	<_Y1>	REAL	2. позиция Y
7	X2	<_X2>	REAL	3. позиция X
8	Y2	<_Y2>	REAL	3. позиция Y
9	X3	<_X3>	REAL	4. позиция X
10	Y3	<_Y3>	REAL	4. позиция Y
11	X4	<_X4>	REAL	5. позиция X
12	Y4	<_Y4>	REAL	5. позиция Y
13	X5	<_X5>	REAL	6. позиция X
14	Y5	<_Y5>	REAL	6. позиция Y
15	X6	<_X6>	REAL	7. позиция X
16	Y6	<_Y6>	REAL	7. позиция Y
17	X7	<_X7>	REAL	8. позиция X
18	Y7	<_Y7>	REAL	8. позиция Y
19	X8	<_X8>	REAL	9. позиция X
20	Y8	<_Y8>	REAL	9. позиция Y

№	Параметр в маске	Параметр внутр.	Тип данных	Значение
21		<_VARI>	INT	Обработка
				ТРЕТЬЯ ПОЗИЦИЯ:
				(только для вызова из Jobshop) (пока обрабатываются только 0 и 2)
				0 = нет зажима шпинделя
				1 = зажим шпинделя только при вертикальном врезании с G00 или G01
				2 = зажим шпинделя во время всей обработки
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: зарезервировано
				ПЯТАЯ ПОЗИЦИЯ: образец позиции с круговой осью/без нее – комбинация осей (с <_VARI> ШЕСТАЯ ПОЗИЦИЯ)
				0 = ХУ (только ХУ без круговой оси, совместимость)
				1 = Х, У или Z и круговая ось: ХА, УВ, ZС (1 круговая ось с гео-осью, вокруг которой вращается круговая ось)
				2 = ХУ и круговая ось: ХУА, ХУВ, ХУС (1 круговая ось с 1-й и 2-й гео-осями, без TRACYL)
				ШЕСТАЯ ПОЗИЦИЯ: Круговая ось
				0 = без круговой оси (только ХУ, совместимость)
				1 = ось А (круговая ось вокруг Х)
				2 = ось В (круговая ось вокруг У)
				3 = ось С (круговая ось вокруг Z)
				ВОСЬМАЯ ПОЗИЦИЯ + СЕДЬМАЯ ПОЗИЦИЯ: образец позиции с круговой осью – смещение (при нескольких круговых осях вокруг одной и той же оси; если индекс слишком велик, то 1-й оси)
				00 = 1-я ось А, В или С или при совместимости
				01 = 2-я ось А, В или С
				...
				19 = 20-я ось А, В или С
22		<_UMODE>	INT	Выбор зажимаемого шпинделя: (только для вызова из Jobshop) (вызов цикла пользователя CUST_TECHCYC)
				3 = зажим/отпускание главного шпинделя
				23 = зажим/отпускание противощпинделя

№	Параметр в маске	Параметр внутр.	Тип данных	Значение
23		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
24		<S_ABA>	INT	Альтернатива для всех АВ-позиций (9-значное десятичное значение) Число разрядов: 876543210 (разряд соответствует позиции ABn)
				Значение разряда:
				1 = абсолютное (1-я запрограммированная позиция всегда абсолютная!)
25	A0	<S_AB0>	REAL	2 = инкрементальное
26	A1	<S_AB1>	REAL	1-я позиция круговой оси при образце позиции с круговой осью (в комбинации с <_VARI>)
27	A2	<S_AB2>	REAL	2-я позиция круговой оси при образце позиции с круговой осью
28	A3	<S_AB3>	REAL	3-я позиция круговой оси при образце позиции с круговой осью
29	A4	<S_AB4>	REAL	4-я позиция круговой оси при образце позиции с круговой осью
30	A5	<S_AB5>	REAL	5-я позиция круговой оси при образце позиции с круговой осью
31	A6	<S_AB6>	REAL	6-я позиция круговой оси при образце позиции с круговой осью
32	A7	<S_AB7>	REAL	7-я позиция круговой оси при образце позиции с круговой осью
33	A8	<S_AB8>	REAL	8-я позиция круговой оси при образце позиции с круговой осью

Примечание

Ненужные позиции параметров X1/Y1/A1 — X8/Y8/A8 можно пропустить. Но альтернативные значения для <_XA>, <_YA> и <S_ABA> должны быть указаны полностью для всех 9 позиций.

При образцах позиции XA, YB или ZC (одна гео-ось и круговая ось) ось плоскости обработки, которая не перемещается через образец позиции (при G17 и XA это ось Y), нужно позиционировать перед вызовом цикла.

20.1.36 CYCLE830 - Глубокое сверление 2**Синтаксис**

```
CYCLE830(<RTP>, <RFP>, <SDIS>, <_DP>, <FDEP>, <_DAM>, <DTB>, <DTS>,
<FRF>, <VARI>, <_MDEP>, <_VRT>, <_DTD>, <_DIS1>, <S_FP>, <S_SDAC2>,
<S_SV2>, <S_FB>, <_SDAC>, <_SV1>, <S_SPOS>, <S_ZA>, <S_FA>, <S_ZP>,
<S_FS>, <S_ZS1>, <S_ZS2>, <S_N>, <S_ZD>, <S_FD>, <S_FR>, <S_SDAC3>,
```

<S_SV3>, <S_CON>, <S_COFF>, <_GMODE>, <_DMODE>, <_AMODE>,
 <S_AMODE2>, <S_AMODE3>, <S_ZPV>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<RFP>	REAL	Опорная точка (абс.)
3	sc	<SDIS>	REAL	Безопасное расстояние (аддитивно к исходной точке, без знака)
4	Z1	<_DP>	REAL	Конечная глубина сверления абс./инкр. (см. <_AMODE> ПЕРВАЯ ПОЗИЦИЯ)
5	D	<FDEP>	REAL	1. Глубина сверления для ломки/удаления стружки, абсолютная или инкрементальная, относительно исходной точки при засверливании/без засверливания или относительно глубины базового отверстия <_AMODE> ПЯТАЯ ПОЗИЦИЯ)
6	DF	<_DAM>	REAL	Величина/процент для каждой следующей подачи, величина депрессии или процент, (см. <_AMODE> ШЕСТАЯ ПОЗИЦИЯ)
7	DTB	<DTB>	REAL	Время ожидания на каждой глубине сверления (см. <_AMODE> ВТОРАЯ ПОЗИЦИЯ)
8	DTS	<DTS>	REAL	Время ожидания при удалении стружки в начальной точке (см. <_AMODE> ТРЕТЬЯ ПОЗИЦИЯ)
9	FD1	<FRF>	REAL	Процент для подачи при первой подаче, см. (см. <_AMODE> ВОСЬМАЯ ПОЗИЦИЯ)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
10		<VARI>	INT	Обработка		
				ПЕРВАЯ ПОЗИЦИЯ:	ломка стружки/удаление стружки	
					0 =	в одном проходе
					1 =	ломка стружки
					2 =	удаление стружки
					3 =	ломка стружки и удаление стружки
				ВТОРАЯ ПОЗИЦИЯ:	Отвод при удалении стружки	
					0 =	на глубину базового отверстия
					1 =	на безопасное расстояние
				ТРЕТЬЯ ПОЗИЦИЯ:	Мягкое врезание	
					0 =	нет
					1 =	да
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	Сквозное сверление	
					0 =	нет
					1 =	да
				ПЯТАЯ ПОЗИЦИЯ:	Засверливание/базовое отверстие	
0 =	без засверливания					
1 =	с засверливанием					
2 =	с базовым отверстием					
ШЕСТАЯ ПОЗИЦИЯ:	Отвод					
	0 =	на глубину базового отверстия				
	1 =	на плоскость отвода				
11	V1	<_MDEP>	REAL	Мин. подача на глубину инкрементальная (только при проценте для депрессии)		
12	V2	<_VRT>	REAL	Величина отвода после каждой обработки инкрементальная (только при ломке стружки)		
					0 =	стандартное значение 1 мм
				> 0 =	переменная величина отвода	
13	DT	<_DTD>	REAL	Время ожидания на конечной глубине сверления, см. <_AMODE> ЧЕТВЕРТАЯ ПОЗИЦИЯ)		
14	V3	<_DIS1>	REAL	Упреждающий зазор инкрементальный, только при удалении стружки (см. <_AMODE> СЕДЬМАЯ ПОЗИЦИЯ)		
15	FP	<S_FP>	REAL	Подача для ввода в базовое отверстие, как значение или в % (в комбинации с <S_ AMODE2> ТРЕТЬЯ ПОЗИЦИЯ)		
16		<S_SDAC2>	INT	Направление вращения шпинделя при подводе		
					3 =	M3
					4 =	M4
					5 =	M5 (по умолчанию)
17	SP	<S_SV2>	REAL	Подвод с	постоянной скоростью шпинделя (см. <S_ AMODE2> ВОСЬМАЯ ПОЗИЦИЯ)	
	V4				постоянной скоростью резания	
					Скорость шпинделя в % от скорости сверления	

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
18	F	<S_FB>	REAL	Подача сверления (см. <S_AMODE2> ПЕРВАЯ ПОЗИЦИЯ)
19		<_SDAC>	REAL	Направление вращения шпинделя при сверлении
				3 = M3
				4 = M4
20	S	<_SV1>	REAL	Сверление с
	V5			постоянной скоростью шпинделя (см. <S_AMODE2> СЕДЬМАЯ ПОЗИЦИЯ)
				постоянной скоростью резания
21	SPOS	<S_SPOS>	REAL	Позиция шпинделя, только при подводе с M5
22	ZA	<S_ZA>	REAL	Глубина засверливания инкрементальная относительно исходной точки или абсолютная (см. <S_AMODE3> ПЕРВАЯ ПОЗИЦИЯ)
23	FA	<S_FA>	REAL	Подача засверливания как значение или в % (в комбинации с <S_AMODE2> ВТОРАЯ ПОЗИЦИЯ)
24	ZP	<S_ZP>	REAL	Базовое отверстие инкрементальное относительно исходной точки, абсолютное или как коэффициент диаметра сверла (см. <S_AMODE3> ВТОРАЯ ПОЗИЦИЯ)
25	FS	<S_FS>	REAL	Подача врезания как значение или в % (в комбинации с <S_AMODE2> ЧЕТВЕРТАЯ ПОЗИЦИЯ)
26	ZS1	<S_ZS1>	REAL	Глубина каждого врезания с постоянной подачей (инкр.)
27	ZS2	<S_ZS2>	REAL	Глубина каждого врезания для увеличения подачи (инкр.)
28	n	<S_N>	INT	Число ходов стружколомания перед каждым удалением стружки
29	ZD	<S_ZD>	REAL	Остаточная глубина сверления инкрементальная относительно конечной глубины сверления или абсолютная (в комбинации с <S_AMODE3> ТРЕТЬЯ ПОЗИЦИЯ)
30	FD	<S_FD>	REAL	Остаточная подача сверления как значение или в % (в комбинации с <S_AMODE2> ПЯТАЯ ПОЗИЦИЯ)
31	FR	<S_FR>	REAL	Подача при отводе (в комбинации с <S_AMODE2> ШЕСТАЯ ПОЗИЦИЯ)
32		<S_SDAC3>	INT	Направление вращения шпинделя при отводе
				3 = M3
				4 = M4
				5 = M5
33	SR	<S_SV3>	REAL	Отвод с
	V6			постоянной скоростью шпинделя (см. <S_AMODE2> ДЕВЯТАЯ ПОЗИЦИЯ)
				постоянной скоростью резания
				Скорость шпинделя в % от скорости сверления
34	СОЖ вкл	<S_CON>	STRING[10]	СОЖ вкл, М-команда или вызов подпрограммы
35	СОЖ выкл	<S_COFF>	STRING[10]	СОЖ выкл, М-команда или вызов подпрограммы
36		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных значений)
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: глубина сверления относительно острия/хвостовика
				0 = Острие
				1 = Хвостовик

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
37		<_DMODE>	INT	Режим индикации	
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19
					0 = совместимость, остается активной действующая перед вызовом цикла плоскость
					1 = G17 (активна только в цикле)
					2 = G18 (активна только в цикле)
					3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ:	зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	зарезервировано
				ПЯТАЯ ПОЗИЦИЯ:	Технологическое масштабирование внутри масок цикла (Страница 870)
					0 = Ввод: полный
					1 = Ввод: простой

№	Параметр в маске	Параметр внутренний	Тип данных	Значение			
38		<_AMODE>	INT	Альтернативный режим 1			
				ПЕРВАЯ ПОЗИЦИЯ:	глубина сверления = конечная глубина сверления Z1 абс./инкр.		
					0 =	инкрементальная	
					1 =	абсолютная	
				ВТОРАЯ ПОЗИЦИЯ:	время ожидания на каждой глубине сверления DTB в секундах/оборотах		
					0 =	в секундах	
					1 =	в оборотах	
				ТРЕТЬЯ ПОЗИЦИЯ:	Время ожидания для удаления стружки DTS в секундах/оборотах		
					0 =	в секундах	
					1 =	в оборотах	
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	время ожидания на конечной глубине сверления DT в секундах/оборотах		
					0 =	в секундах	
					1 =	в оборотах	
				ПЯТАЯ ПОЗИЦИЯ:	1. Глубина сверления D (абс./инкр.)		
					0 =	инкрементальная	
					1 =	абсолютная	
				ШЕСТАЯ ПОЗИЦИЯ:	величина/процент DF для каждой следующей подачи (депрессия)		
					0 =	Значение	
					1 =	процент (от 0,001 до 100 %)	
				СЕДЬМАЯ ПОЗИЦИЯ:	упреждающий зазор V3 автоматический/ручной		
					0 =	автоматический (вычисляется в цикле)	
					1 =	ручной (запрограммированное значение)	

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
39		<S_AMODE2 >	INT	Альтернативный режим 2
				ПЕРВАЯ ПОЗИЦИЯ:
				ПЕРВАЯ ПОЗИЦИЯ: Подача сверления F
				0 = F/мин
				1 = F/об
				ВТОРАЯ ПОЗИЦИЯ:
				нормирование подачи засверливания FA
				0 = в % от подачи сверления
				1 = F/мин
				2 = F/об
				ТРЕТЬЯ ПОЗИЦИЯ:
				нормирование подачи для ввода в базовое отверстие FP
				0 = в % от подачи сверления
				1 = F/мин
				2 = F/об
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				нормирование подачи врезания FS
				0 = в % от подачи сверления
				1 = F/мин
				2 = F/об
				ПЯТАЯ ПОЗИЦИЯ:
				нормирование подачи сквозного сверления FD
				0 = в % от подачи сверления
				1 = F/мин
				2 = F/об
				ШЕСТАЯ ПОЗИЦИЯ:
				Подача при отводе FR
				0 = F/мин
				1 = Ускоренный ход
				СЕДЬМАЯ ПОЗИЦИЯ:
				сверление - скорость шпинделя/скорость резания (S/V5)
				0 = постоянная скорость шпинделя
				1 = постоянная скорость резания
				ВОСЬМАЯ ПОЗИЦИЯ:
				подвод со скоростью шпинделя/скоростью резания (SP/V4)
				0 = постоянная скорость шпинделя
				1 = постоянная скорость резания
				2 = скорость шпинделя в % от скорости сверления
				ДЕВЯТАЯ ПОЗИЦИЯ:
				отвод - скорость шпинделя/скорость резания (SR/V6)
				0 = постоянная скорость шпинделя
				1 = постоянная скорость резания
				2 = скорость шпинделя в % от скорости сверления

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
40		<S_АМОДЕЗ>	INT	Альтернативный режим 3		
				ПЕРВАЯ ПОЗИЦИЯ:	глубина засверливания ZA абс./инкр.	
					0 =	инкрементальная
					1 =	абсолютная
				ВТОРАЯ ПОЗИЦИЯ:	глубина базового отверстия ZP	
					0 =	инкрементальная
					1 =	абсолютная
				ТРЕТЬЯ ПОЗИЦИЯ:	коэффициент диаметра сверла	
					остаточная глубина сверления ZD инкр./абс.	
					0 =	инкрементальная
1 =	абсолютная					
41	ZPV	<S_ZPV>	REAL	Упреждающий зазор, инкрементальный, на глубину базового отверстия		

20.1.37 CYCLE832 - High Speed Settings

Синтаксис

CYCLE832 (<S_TOL>, <S_TOLM>, <S_OTOL>)

Примечание

CYCLE832 не освобождает изготовителя станка от необходимости выполнения задач по оптимизации при вводе станка в эксплуатацию. Это относится к оптимизации участвующих в обработке осей и настройкам NCU (предупреждение, ограничение рывка и т.д.).

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
1	Допуск	<S_TOL>	REAL	Допуск контура Допуск контура соответствует осевому допуску гео-осей.
2		<S_TOLM>	INT	Режим обработки (технология) ПЕРВАЯ ПОЗИЦИЯ: 0 = Отмена 1 = чистовая обработка (Finish) 2 = получистовая обработка (Semifinish) 3 = черновая обработка (Rough) ВТОРАЯ ПОЗИЦИЯ: 0 = совместимость¹⁾ или нет допуска ориентации 1 = Допуск ориентации в параметре <S_OTOL> ТРЕТЬЯ ... СТОТЫСЯЧНАЯ занято по причине Совместимость СЕДЬМАЯ ПОЗИЦИЯ: 0 = Совместимость. Автоматически используется наилучшая из доступных функций изготовления пресс-форм: - Опция Top Surface не активна: ⇒ Advanced Surface - Опция Top Surface активна: ⇒ Top Surface со сглаживанием 1 = Top Surface без сглаживания 2 = Top Surface со сглаживанием
3	ORI-допуск	<S_OTOL>	REAL	Допуск ориентации или обозначение версии CYCLE832 Параметр допуска для ориентации инструмента. Необходим для выполнения высокоскоростной программы обработки на станках с динамической трансформацией ориентации (например, 5-осевая обработка). Параметр <S_OTOL> нужно запрограммировать. Это же относится и к приложениям на 3-осевых станках для программ без ориентации инструмента (<S_OTOL> = 1).

¹⁾ Допуск ориентации, выведенный из допуска контура, умноженного на коэффициент из установочных данных циклов SD55441 до SD55443.

Литература:

Руководство по вводу в эксплуатацию - Базовое ПО и ПО управления; SINUMERIK Operate (IM9), глава "Конфигурирование функции High Speed Setting (CYCLE832)"

Ввод открытым текстом

Для улучшения читабельности вызова цикла, параметр <S_TOLM> (Режим обработки) может быть введен и открытым текстом. Открытые тексты не имеют языковой зависимости. Могут вводиться следующие данные:

_OFF	Для	0	Отмена
_FINISH	Для	1	Чистовая обработка
_SEMIFIN	Для	2	Предварительная чистовая обработка
_ROUGH	Для	3	Черновая обработка
_ORI_FINISH	Для	11	Получистовая обработка с вводом допуска ориентации
_ORI_SEMIFIN	Для	12	Предварительная чистовая обработка с вводом допуска ориентации
_ORI_ROUGH	Для	13	Черновая обработка с вводом допуска ориентации
_TOP_SURFACE_SMOOTH_OFF	Для	1000000	Top Surface без сглаживания
_TOP_SURFACE_SMOOTH_ON	Для	2000000	Top Surface со сглаживанием

Пример составления открытого текста, вводимого в Top Surface, приведен ниже:

```
CYCLE832 (0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

Примечание

Открытые тексты взяты из имен группы G-функций 59 (динамический режим для интерполяции траектории). Этими открытыми текстами при использовании четко различаются 3-осевые станки и станки с многоосевой трансформацией ориентации (TRAORI).

Сброс CYCLE832

При отмене CYCLE832 параметр <S_TOL> должен передаваться с нулем.

Пример: CYCLE832 (0, 0, 1)

Синтаксис CYCLE832 () также разрешен для отмены CYCLE832.

Примеры

Пример 1: CYCLE832 на 3-осевом станке без трансформации ориентации**а) Вызов цикла с вводом открытого текста**

Программный код	Комментарий
G710	; Метрическая система единиц.
CYCLE832 (0.004, _FINISH, 1)	; Вызов CYCLE832 с: допуск контура = 0,004 мм, режим обработки: чистовая обработка
...	; Выполнение программы высокоскоростной обработки

б) Вызов цикла без ввода открытого текста

Программный код	Комментарий
G710	; см. выше
CYCLE832 (0.004, 1, 1)	; см. выше
...	; см. выше

Пример 2: CYCLE832 на 5-осевом станке с трансформацией ориентации**а) Вызов цикла и отмена с вводом открытого текста**

Программный код	Комментарий
G710	; Метрическая система единиц.
TRAORI	; Активировать трансформацию ориентации.
CYCLE832 (0.3, _ORI_ROUGH, 0.8)	; Вызов CYCLE832 с: допуск контура = 0,3 мм, режим обработки: черновая обработка с вводом допуска ориентации, допуск ориентации = 0,8 градуса
...	; Выполнение программы высокоскоростной обработки
CYCLE832 (0, _OFF, 1)	; Допуск контура = 0, режим обработки: Отмена CYCLE832, допуск ориентации = 0 градусов

б) Вызов цикла и отмена без ввода открытого текста

Программный код	Комментарий
G710	; см. выше
TRAORI	; см. выше
CYCLE832 (0.3, 13, 0.8)	; см. выше
...	; см. выше
CYCLE832 (0, 0, 1)	; см. выше

20.1.38 CYCLE840 - Нарезание внутренней резьбы с компенсирующим патроном

Синтаксис

CYCLE840 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>, <ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение			
1	RP	<RTP>	REAL	Плоскость отвода (абс.)			
2	Z0	<RFP>	REAL	Исходная точка (абс.)			
3	sc	<SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)			
4	Z1	<DP>	REAL	Глубина сверления (абс.), см. <_AMODE>			
5	Z1	<DPR>	REAL	Глубина сверления (инкр.), см. <_AMODE>			
6	DT	<DTB>	REAL	Время ожидания на глубине сверления/на безопасном расстоянии после отвода в секундах, см. <ENC>			
7		<SDR>	INT	Направление вращения для отвода			
8	SDE	<SDAC>	INT	Направление вращения после завершения цикла			
9		<ENC>	INT	Нарезание внутренней резьбы с датчиком шпинделя (G33)/нарезание внутренней резьбы без датчика шпинделя (G63)			
					0 =	с датчиком шпинделя	- шаг из <MPIT>/<PIT> - без DT
					20 =	с датчиком шпинделя	- шаг из <MPIT>/<PIT> - с DT после отвода на безопасное расстояние
					11 =	без датчика шпинделя	- шаг из <MPIT>/<PIT> - с DT на глубине сверления
					1 =	без датчика шпинделя	- шаг из запрограммированной подачи - с DT на глубине сверления (подача = скорость · шаг)
10		<MPIT>	REAL	Размер резьбы только для "ISO метрической" (шаг вычисляется при внутренней обработке) Диапазон значений: 3 до 48 (для M3 до M48), как альтернатива <PIT>			
11		<PIT>	REAL	Шаг резьбы как значение, единицу см. <_PITA> Диапазон значений: > 0, как альтернатива MPIT			

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Значение			
12		<_AXN>	INT	Ось сверления	0 =	3-я гео-ось	
					1 =	1-я гео-ось	
					2 =	2-я гео-ось	
					≥ 3 =	3-я гео-ось	
13		<_PITA>	INT	Единица измерения шага резьбы (обработка <PIT> и <MPIT>)			
					0 =	шаг в мм	- обработка <MPIT>/<PIT>
					1 =	шаг в мм	- обработка <PIT>
					2 =	шаг в TPI	- обработка <PIT> (витков резьбы на дюйм)
					3 =	шаг в дюймах	- обработка <PIT>
					4 =	MODUL	- обработка <PIT>
14	<_TECHNO>	INT	Технология ¹⁾				
			ПЕРВАЯ ПОЗИЦИЯ:	параметры точного останова			
				0 =	активны параметры точного останова как перед вызовом цикла		
				1 =	точный останов G601		
				2 =	точный останов G602		
				3 =	точный останов G603		
			ВТОРАЯ ПОЗИЦИЯ:	предуправление			
				0 =	активно с/без предупреждения как перед вызовом цикла		
				1 =	с предупреждением FFWON		
				2 =	без предупреждения FFWOF		
15		<_PITM>	STRING[15]	Строка как меркер для ввода шага резьбы ²⁾			
16		<_PTAB>	STRING[5]	Строка для таблицы резьб ("", "ISO", "BSW", "BSP", "UNC") ²⁾			
17		<_PTABA>	STRING[20]	Строка для выбора в таблице резьб (к примеру, "M 10", "M 12", ...) ²⁾			
18		<_GMODE>	INT	зарезервировано			

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
19		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:	зарезервировано	
				ТРЕТЬЯ ПОЗИЦИЯ:	зарезервировано	
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	режим совместимости (только для маски ввода для обратного перевода), если MD 52216 Бит0 = 1 ¹⁾	
					0 =	технологические параметры отображаются (совместимость): действуют TECHNO-параметры
					1 =	технологические параметры не отображаются: действует технология "как запрограммировано перед вызовом цикла"
				ПЯТАЯ ПОЗИЦИЯ:	Технологическое масштабирование внутри масок цикла (Страница 870)	
					0 =	Ввод: полный
1 =	Ввод: простой					
20		<_AMODE>	INT	Альтернативный режим		
				ПЕРВАЯ ПОЗИЦИЯ:	глубина сверления Z1 (абс./инкр.)	
					0 =	совместимость, из программирования <DP>/<DPR>
					1 =	инкрементальная
					2 =	абсолютная

¹⁾ Поля технологии могут быть скрыты в зависимости от установочных данных SD52216 MCS_FUNCTION_MASK_DRILL

²⁾ Параметры 15, 16 и 17 используются только при выборе резьбы в таблицах резьб маски ввода. Доступ к таблицам резьб через определение цикла при выполнении цикла невозможен!

20.1.39 CYCLE899 - фрезерование открытого паза

Синтаксис

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>, <_CDIR>,
<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>, <_ZFS>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	RP	<_RTP>	REAL	Плоскость отвода (абс.)
2	Z0	<_RFP>	REAL	Исходная точка оси инструмента (абс.)
3	sc	<_SDIS>	REAL	Безопасное расстояние (вводится аддитивно к исходной точке, без знака)
4	Z1	<_DP>	REAL	Глубина паза (абс./инкр.), см.<_AMODE>
5	I	<_LENG>	REAL	Длина канавки (инкр.)
6	W	<_WID>	REAL	Ширина канавки (инкр.)
7	X0	<_PA>	REAL	Исходная точка/стартовая позиция 1-й оси (абс.)
8	Y0	<_PO>	REAL	Исходная точка/стартовая позиция 2-й оси (абс.)
9	$\alpha 0$	<_STA>	REAL	Угол поворота к 1-й оси
10	DZ	<_MID>	REAL	Макс. глубина подачи (инкр.), только для вихревого фрезерования
11	DX	<_MIDA>	REAL	Макс. подача в плоскости, см. <_AMODE>
12	UX	<_FAL>	REAL	Чистовой припуск плоскость
13	UZ	<_FALD>	REAL	Чистовой припуск глубина
14	F	<_FFP1>	REAL	Подача
15		<_CDIR>	INT	Направление фрезерования
				ПЕРВАЯ ПОЗИЦИЯ:
				0 = Синхронный ход
				1 = Противоход
				4 = Попеременно
16		<_VARI>	INT	Обработка
				ПЕРВАЯ ПОЗИЦИЯ:
				1 = Черновая обработка
				2 = Чистовая обработка
				3 = Чистовая обработка основания
				4 = Чистовая обработка края
				5 = Предварительная чистовая обработка
				6 = Снятие фаски
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				1 = Вихревое фрезерование
				2 = Врезное фрезерование

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
17		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)
				ПЕРВАЯ ПОЗИЦИЯ: зарезервировано
				ВТОРАЯ ПОЗИЦИЯ: зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: выбор обработки/только расчет стартовой точки
				1 = обычная обработка
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: указание размеров через центр / кромку
				0 = указание размеров через центр
				1 = указание размеров через кромку "слева" ("-" направление 1-й оси)
18		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ: плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ: --- зарезервировано
				ТРЕТЬЯ ПОЗИЦИЯ: --- зарезервировано
				ЧЕТВЕРТАЯ ПОЗИЦИЯ: --- зарезервировано
				ПЯТАЯ ПОЗИЦИЯ: Технологическое масштабирование внутри масок цикла (Страница 870)
19		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ: глубина канавки Z1
				0 = абсолютная
				1 = инкрементальная
				ВТОРАЯ ПОЗИЦИЯ: единица для подачи в плоскости (<_MIDA>) DXY
				0 = мм
				1 = % от диаметра инструмента
				ТРЕТЬЯ ПОЗИЦИЯ: глубина врезания при снятии фаски ZFS
20		<_UMODE>	INT	0 = абсолютная
				1 = инкрементальная
				зарезервировано

20.1 Технологические циклы

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
21	FS	<_FS>	REAL	Ширина фаски (инкр.)
22	ZFS	<_ZFS>	REAL	Глубина врезания (острие инструмента) при снятии фаски (абс./инкр.) см. <_AMODE>

20.1.40 CYCLE930 - выточка

Синтаксис

CYCLE930(<_SPD>, <_SPL>, <_WIDG>, <_WIDG2>, <_DIAG>, <_DIAG2>, <_STA>, <_ANG1>, <_ANG2>, <_RCO1>, <_RCI1>, <_RCI2>, <_RCO2>, <_FAL>, <_IDEP1>, <_SDIS>, <_VARI>, <_DN>, <_NUM>, <_DBH>, <_FF1>, <_NR>, <_FALX>, <_FALZ>, <_DMODE>, <_AMODE>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	X0	<_SPD>	REAL	Исходная точка в поперечной оси (всегда диаметр)
2	Z0	<_SPL>	REAL	Исходная точка в продольной оси
3	B1	<_WIDG>	REAL	Ширина выточки внизу
4	B2	<_WIDG2>	REAL	Ширина выточки сверху (только для интерфейса)
5	T1	<_DIAG>	REAL	Глубина выточки в исходной точке, при абс. и продольной обработке = диаметр, в остальных случаях инкр.
6	T2	<_DIAG2>	REAL	Глубина выточки напротив исходной точки (только для интерфейса), при абс. и продольной обработке = диаметр, в остальных случаях инкр.
7	α0	<_STA>	REAL	Угол наклона ($-180 \leq \text{<_STA>} \leq 180$)
8	α1	<_ANG1>	REAL	Угол профиля 1 ($0 \leq \text{<_ANG1>} < 90$) на определенной через исходную точку стороне выточки
9	α2	<_ANG2>	REAL	Угол профиля 2 ($0 \leq \text{<_ANG2>} < 90$) напротив исходной точки
10	R1/FS1	<_RCO1>	REAL	Радиус закругления или ширина фаски 1, снаружи на исходной точке
11	R2/FS2	<_RCI1>	REAL	Радиус закругления или ширина фаски 2, внутри на исходной точке
12	R3/FS3	<_RCI2>	REAL	Радиус закругления или ширина фаски 3, внутри напротив исходной точки
13	R4/FS4	<_RCO2>	REAL	Радиус закругления или ширина фаски 4, снаружи напротив исходной точки
14	U	<_FAL>	REAL	Чистовой припуск в X и Z, см. <_VARI>(ПЯТАЯ ПОЗИЦИЯ) (вводится без знака)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
15	D	<_IDEP1>	REAL	Макс. подача на глубину при врезании (вводится без знака)		
					0 =	1. проход резца прямо на всю глубину
					> 0 =	1-й проход <_IDEP1>, 2-й проход $2 \cdot <_IDEP1>$ и т.д.
16	sc	<_SDIS>	REAL	Безопасное расстояние (вводится без знака)		
17		<_VARI>	INT	Режим обработки		
				ПЕРВАЯ ПОЗИЦИЯ:	зарезервировано	
				ВТОРАЯ ПОЗИЦИЯ:	технологическая обработка	
					1 =	черновая обработка
					2 =	чистовая обработка
					3 =	черновая и чистовая обработка
				ТРЕТЬЯ ПОЗИЦИЯ:	положение продольное/поперечное снаружи/внутри +Z/+Z или +X/-X	
					1 =	продольное/снаружи +Z
					2 =	поперечное/внутри -X
					3 =	продольное/внутри +Z
					4 =	поперечное/внутри +X
					5 =	продольное/снаружи -Z
					6 =	поперечное/снаружи -X
					7 =	продольное/внутри -Z
					8 =	поперечное/снаружи +X
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	положение исходной точки	
					0 =	исходная точка сверху
					1 =	исходная точка снизу
				ПЯТАЯ ПОЗИЦИЯ:	определение действия чистовых припусков	
					0 =	чистовой припуск U параллельно контуру
					1 =	чистовые припуски UX и UZ отдельно
18		<_DN>	INT	D-номер для 2-го резца инструмента		
					> 0 =	D-номер для коррекции инструмента 2-го резца прорезного инструмента
					0 =	2-й резец не запрограммирован
19	n	<_NUM>	INT	Число выточек (0 = 1 выточка)		
20	DP	<_DBH>	REAL	Расстояние между выточками (необходимо, только если <_NUM>> 1)		
21	F	<_FF1>	REAL	Подача		

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
22		<_NR>	INT	Идентификатор для формы выточки соответствует вертикальным программным клавишам для выбора формы		
					0 =	боковые поверхности 90° без фасок/закруглений
					1 =	наклонные боковые стороны с фасками/закруглениями (без α0)
					2 =	как 1, но на конусе (с α0)
23	UX	<_FALX>	REAL	Чистовой припуск в оси X, см. <_VARI>(ПЯТАЯ ПОЗИЦИЯ) (вводится без знака)		
24	UZ	<_FALZ>	REAL	Чистовой припуск в оси Z, см. <_VARI>(ПЯТАЯ ПОЗИЦИЯ) (вводится без знака)		
25		<_DMODE>	INT	Режим индикации		
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19	
					0 =	совместимость, остается активной действующая перед вызовом цикла плоскость
					1 =	G17 (активна только в цикле)
					2 =	G18 (активна только в цикле)
					3 =	G19 (активна только в цикле)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
26		<_AMODE>	INT	Альтернативный режим
				ПЕРВАЯ ПОЗИЦИЯ:
				указание размеров Глубина (только для интерфейса)
				0 = на исходной точке
				1 = напротив исходной точки
				ВТОРАЯ ПОЗИЦИЯ:
				глубина
				0 = абсолютная
				1 = инкрементальная
				ТРЕТЬЯ ПОЗИЦИЯ:
				указание размеров Ширина (только для интерфейса)
				0 = на наружном диаметре (сверху)
				1 = на внутреннем диаметре (снизу)
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				радиус/фаска 1 (<_RCO1>)
				0 = радиус
				1 = фаска
				ПЯТАЯ ПОЗИЦИЯ:
				радиус/фаска 2 (<_RCI1>)
				0 = радиус
				1 = фаска
				ШЕСТАЯ ПОЗИЦИЯ:
				радиус/фаска 3 (<_RCI2>)
				0 = радиус
				1 = фаска
				СЕДЬМАЯ ПОЗИЦИЯ:
				радиус/фаска 4 (<_RCO2>)
				0 = радиус
				1 = фаска

20.1.41 CYCLE940 - формирование канавки

С помощью цикла CYCLE940 можно запрограммировать различные канавки. Частично они очень сильно различаются в параметрировании.

Дополнительные колонки в таблице показывают, какой параметр для какой формы канавки необходим. Они соответствуют вертикальным программным клавишам выбора в маске цикла:

- E: канавка формы E
- F: канавка формы F
- A-D: резьбовая канавка DIN (формы A-D)
- t: резьбовая канавка (свободное определение формы)

Синтаксис

CYCLE940 (<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>, <_VARI>,
 <_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>, <_FAL>,
 <_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>)

Параметр

№	Пара-метр в маске	Параметр внутрен-ний	Тип дан-ных	Прогр. для фор-мы				Объяснение			
				E	F	A-D	T				
1	X0	<_SPD>	REAL	x	x	x	x	Исходная точка в поперечной оси (всегда диаметр)			
2	Z0	<_SPL>	REAL	x	x	x	x	Исходная точка продольной оси (абс.)			
3	FORM	<_FORM>	CHAR	x	x	x	x	Форма канавки (прописные буквы, например, "T") Выбор, из какой таблицы должны быть взяты значения ка-навки			
									A =	снаружи, источник DIN76, A = обычная	
									B =	снаружи, источник DIN76, B = короткая	
									C =	внутри, источник DIN76, C = обычная	
									D =	внутри, источник DIN76, D = короткая	
									E =	источник DIN509	
									f =	источник DIN509	
									T=	свободная форма	
4	LAGE	<_LAGE>	INT	x	x	x	x	Положение канав-ки (параллельно Z)	0 =	снаружи +Z: ____	
									1 =	снаружи -Z: ____/	
									2 =	внутри +Z: /-----	
									3 =	внутри -Z: -----\	
5	sc	<_SDIS>		x	x	x	x	Безопасное расстояние (инкр.)			
6	F	<_FFP>		x	x	x	x	Подача обработки (мм/об)			
7		<_VARI>	INT	-	-	x	x	Режим обработки			
									ПЕРВАЯ ПОЗИ-ЦИЯ:	обработка	
										1 =	черновая обработка
										2 =	чистовая обработка
										3 =	черновая + чистовая
									ВТОРАЯ ПОЗИ-ЦИЯ:	стратегия обработки	
										0 =	параллельно контуру
										1 =	продольная
								Канавки формы E и F всегда обрабатываются за один проход как при чистовой обработке.			
8	X1	<_EPD>		x	x	-	-	Припуск X (абс./инкр.), см. <_AMODE>			
				-	-	-	x	Глубина канавки (абс./инкр.), см. <_AMODE>			

№	Пара-метр в маске	Параметр внутрен-ний	Тип дан-ных	Прогр. для фор-мы				Объяснение			
9	Z1	<_EPL>		-	x	-	-	Припуск Z			
				-	-	-	x	Ширина канавки (абс./инкр.), см. <_AMODE>			
10	R1	<_R1>		-	-	-	x	Радиус закругления на наклоне			
11	R2	<_R2>		-	-	-	x	Радиус закругления в углу			
12	α	<_STA>		-	-	x	x	Угол врезания			
13	VX	<_VRT>		x	x	-	-	Поперечная подача X (абс./инкр.), см. <_AMODE>			
				-	-	x	x	Поперечная подача X при чистовой обработке, (абс./инкр.), см. <_AMODE>			
14	D	<_MID>		-	-	x	x	Подача на глубину			
15	U	<_FAL>		-	-	x	x	Чистовой припуск параллельно контуру, см. <_AMODE>			
16	UX	<_FALX>		-	-	x	x	Чистовой припуск x			
17	UZ	<_FALZ>		-	-	x	x	Чистовой припуск Z			
18	P	<_PITI>	INT	-	-	x	-	Выбор шага, форма A-D, соответствует M1 ... M68			
								0 = 0.20 1 = 0.25 2 = 0.30 3 = 0.35 4 = 0.40 5 = 0.45	6 = 0.50 7 = 0.60 8 = 0.70 9 = 0.75 10 = 0.80 11 = 1.00	12 = 1.25 13 = 1.50 14 = 1.75 15 = 2.00 16 = 2.50 17 = 3.00	18 = 3.50 19 = 4.00 20 = 4.50 21 = 5.00 22 = 5.50 23 = 6.00
				x	x	-	-	Выбор радиуса/глубины, форма E, F			
								0 = 0.6 · 0.3 1 = 1.0 · 0.4 2 = 1.0 · 0.2 3 = 1.6 · 0.3	4 = 2.5 · 0.4 5 = 4.0 · 0.5 6 = 0.4 · 0.2 7 = 0.6 · 0.2	8 = 0.1 · 0.1 9 = 0.2 · 0.1	
19		<_PTAB>	STRING [5]					Строка для таблицы резьб ("", "ISO", "BSW", "BSP", "UNC") (только для интерфейса)			
20		<_PTABA>	STRING [20]					Строка выбора в таблице значений резьбы (например, "M 10", "M 12", ...) (только для поверхности)			
21		<_DMODE>	INT					Режим индикации			
				x	x	x	x	ПЕРВАЯ ПОЗИ-ЦИЯ:	плоскость обработки G17/G18/G19		
									0 =	совместимость, остается активной действующая перед вызовом цикла плоскость	
									1 =	G17 (активна только в цикле)	
									2 =	G18 (активна только в цикле)	
				3 =	G19 (активна только в цикле)						

№	Параметр в маске	Параметр внутренний	Тип данных	Прогр. для формы	Объяснение
22		<_AMODE>	INT	Альтернативный режим	
				x x - x	ПЕРВАЯ ПОЗИЦИЯ:
					Параметр <_EPD> припуск X или глубина канавки
					0 = абсолютная (всегда диаметр)
					1 = инкрементальная
				x x - x	ВТОРАЯ ПОЗИЦИЯ:
					Параметр <_EPL> припуск Z или ширина канавки
					0 = абсолютная
					1 = инкрементальная
				x x x x	ТРЕТЬЯ ПОЗИЦИЯ:
					Параметр <_VRT> поперечная подача X
					0 = абсолютная (всегда диаметр)
					1 = инкрементальная
				- - x x	ЧЕТВЕРТАЯ ПОЗИЦИЯ:
					чистовой припуск
					0 = чистовой припуск параллельно контуру, см. <_FAL>
					1 = чистовой припуск отдельно (<_FALX>/<_FALZ>)

20.1.42 CYCLE951 - обработка резанием

Синтаксис

```
CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>,
<_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>, <_SDIS>,
<_FF1>, <_NR>, <_DMODE>, <_AMODE>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Значение
1	X0	<_SPD>	REAL	Исходная точка (абс., всегда диаметр)
2	Z0	<_SPL>	REAL	Исходная точка (абс.)
3	X1	<_EPD>	REAL	Конечная точка
4	Z1	<_EPL>	REAL	Конечная точка
5	XM α1 α2	<_ZPD>	REAL	Промежуточная точка, см. <_DMODE>(ВТОРАЯ ПОЗИЦИЯ)

№	Параметр в маске	Параметр внутренний	Тип данных	Значение		
6	ZM α1 α2	<_ZPL>	REAL	Промежуточная точка, см. <_DMODE>(ВТОРАЯ ПОЗИЦИЯ)		
7	Положение	<_LAGE>	INT	Положение обрабатываемого угла	0 =	снаружи/сзади
					1 =	снаружи/спереди
					2 =	внутри/сзади
					3 =	внутри/спереди
8	D	<_MID>	REAL	Макс. подача на глубину при врезании		
9	UX	<_FALX>	REAL	Чистовой припуск в X		
10	UZ	<_FALZ>	REAL	Чистовой припуск в Z		
11		<_VARI>	INT	Режим обработки		
				ПЕРВАЯ ПОЗИЦИЯ:	направление резания (продольное или поперечное) в системе координат	
					1 =	продольная
					2 =	поперечная
				ВТОРАЯ ПОЗИЦИЯ:		
					1 =	черновая обработка до чистового припуска
					2 =	чистовая обработка
				ТРЕТЬЯ ПОЗИЦИЯ:	зарезервировано	
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	зарезервировано	
				ПЯТАЯ ПОЗИЦИЯ:	Зарезервировано	
12	R1/FS1	<_RF1>	REAL	Радиус закругления или ширина фаски 1, см. <_AMODE>(ПЯТАЯ ПОЗИЦИЯ)		
13	R2/FS2	<_RF2>	REAL	Радиус закругления или ширина фаски 2, см. <_AMODE>(ШЕСТАЯ ПОЗИЦИЯ)		
14	R3/FS3	<_RF3>	REAL	Радиус закругления или ширина фаски 3, см. <_AMODE> (СЕДЬМАЯ ПОЗИЦИЯ)		
15	sc	<_SDIS>	REAL	Безопасное расстояние		
16	F	<_FF1>	REAL	Подача для черновой/чистовой обработки		
17		<_NR>	INT	Идентификатор для типа резания (соответствует вертикальным программным клавишам для выбора формы):		
					0 =	обработка резаньем 1, угол 90 градусов без фасок/закруглений
					1 =	обработка резаньем 2, угол 90 градусов с фасками/закруглениями
					2 =	обработка резаньем 3, любой угол с фасками/закруглениями

№	Параметр в маске	Параметр внутренний	Тип данных	Значение	
18		<_DMODE>	INT	Режим индикации	
				ПЕРВАЯ ПОЗИЦИЯ:	плоскость обработки G17/G18/G19
					0 = совместимость, остается активной действующая перед вызовом цикла плоскость
					1 = G17 (активна только в цикле)
					2 = G18 (активна только в цикле)
					3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:	форма ввода <_ZPD>/<_ZPL>
					0 = Xm/Zm
					1 = Xm/ α 1
					2 = Xm/ α 2
					3 = α 1/Zm
					4 = α 2/Zm
					5 = α 1/ α 2
21		<_AMODE>	INT	Альтернативный режим	
				ПЕРВАЯ ПОЗИЦИЯ:	промежуточная точка в X
					0 = абсолютная, значение поперечной оси в диаметре
					1 = инкрементальная, значение поперечной оси в радиусе
				ВТОРАЯ ПОЗИЦИЯ:	промежуточная точка в Z
					0 = абсолютная
					1 = инкрементальная
				ТРЕТЬЯ ПОЗИЦИЯ:	конечная точка в X
					0 = абсолютная, значение поперечной оси в диаметре
					1 = инкрементальная, значение поперечной оси в радиусе
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	конечная точка в Z
					0 = абсолютная
					1 = инкрементальная
				ПЯТАЯ ПОЗИЦИЯ:	радиус/фаска 1
					0 = радиус
					1 = фаска
				ШЕСТАЯ ПОЗИЦИЯ:	радиус/фаска 2
					0 = радиус
					1 = фаска
				СЕДЬМАЯ ПОЗИЦИЯ:	радиус/фаска 3
					0 = радиус
					1 = фаска

20.1.43 CYCLE952 - прорезание контура

Синтаксис

```
CYCLE952 (<_PRG>, <_CON>, <_CONR>, <_VARI>, <_F>, <_FR>, <_RP>, <_D>,
<_DX>, <_DZ>, <_UX>, <_UZ>, <_U>, <_U1>, <_BL>, <_XD>, <_ZD>, <_XA>,
<_ZA>, <_XB>, <_ZB>, <_XDA>, <_XDB>, <_N>, <_DP>, <_DI>, <_SC>,
<_DN>, <_GMODE>, <_DMODE>, <_AMODE>, <_PK>, <_DCH>, <_FS>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
1	PRG	<_PRG>	STRING[100]	Имя программы обработки резаньем
2	CON	<_CON>	STRING[100]	Имя программы, из которой считывается обновленный контур заготовки (при обработке остатков материала)
3	CONR	<_CONR>	STRING[100]	Имя программы, в которую записывается обновленный контур заготовки (см. <_AMODE>ПЯТАЯ ПОЗИЦИЯ)

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение	
4		<_VARI>	INT	Режим обработки	
				ПЕРВАЯ ПОЗИЦИЯ:	тип резания
					1 = продольная
					2 = поперечная
					3 = параллельно контуру
				ВТОРАЯ ПОЗИЦИЯ:	технологическая обработка (см. <_GMODE>ТРЕТЬЯ ПОЗИЦИЯ)
					1 = черновая обработка
					2 = чистовая обработка
					3 = зарезервировано
					4 = черновая обработка двухканальная
					5 = чистовая обработка двухканальная
				ТРЕТЬЯ ПОЗИЦИЯ:	Направление обработки
					1 = направление обработки X -
					2 = направление обработки X +
					3 = направление обработки Z -
					4 = направление обработки Z +
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	направление подачи
					1 = снаружи X -
					2 = внутри X +
					3 = торцевая сторона Z -
					4 = задняя сторона Z +
				ПЯТАЯ ПОЗИЦИЯ:	определение действия чистовых припусков
					0 = чистовые припуски UX и UZ отдельно
					1 = чистовой припуск U параллельно контуру
				ШЕСТАЯ ПОЗИЦИЯ:	возврат по контуру
					0 = совместимость, автоматический возврат по контуру
					1 = с возвратом по контуру
					2 = с прямым возвратом
					3 = автоматический возврат по контуру
				СЕДЬМАЯ ПОЗИЦИЯ:	поднутрения
					0 = место не обрабатывается при прорезе, - остаток и токарная выточка, - остаток
					1 = обработка поднутрений
					2 = не обрабатывать поднутрения
				ВОСЬМАЯ ПОЗИЦИЯ:	за/перед центром вращения

№	Параметр в маске	Параметр внутренних	Тип данных	Объяснение		
					0 =	обработка перед центром вращения
					1 =	зарезервировано
5	F	<_F>	REAL	Подача для черновой/чистовой обработки		
	FZ			Подача, абсцисса, токарная выточка		
6	FR	<_FR>	REAL	Подача для врезания в поднутрения при черновой обработке		
	FX			Подача, ордината, токарная выточка		
7	RP	<_RP>	REAL	Плоскость отвода при внутренней обработке (абс., всегда диаметр)		
8	D	<_D>	REAL	Подача черновой обработки, (см. <_AMODE>ПЕРВАЯ ПОЗИЦИЯ)		
9	Dx	<_DX>	REAL	Подача X (см. <_AMODE>ПЕРВАЯ ПОЗИЦИЯ)		
10	DZ	<_DZ>	REAL	Подача Z (см. <_AMODE>ПЕРВАЯ ПОЗИЦИЯ)		
11	UX	<_UX>	REAL	Чистовой припуск X (см. <_VARI>ПЯТАЯ ПОЗИЦИЯ)		
12	UZ	<_UZ>	REAL	Чистовой припуск Z (см. <_VARI> ПЯТАЯ ПОЗИЦИЯ)		
13	U	<_U>	REAL	Чистовой припуск параллельно контуру, см. <_VARI>ПЯТАЯ ПОЗИЦИЯ)		
14	U1	<_U1>	REAL	Дополнительный чистовой припуск при чистовой обработке (см. <_AMODE>ЧЕТВЕРТАЯ ПОЗИЦИЯ)		
15	BL	<_BL>	INT	Определение заготовки	1 =	цилиндр с припуском
					2 =	припуск к контуру готовой детали
					3 =	контур заготовки указан
16	XD	<_XD>	REAL	Определение заготовки X (см. <_AMODE>ШЕСТАЯ ПОЗИЦИЯ)		
17	ZD	<_ZD>	REAL	Определение заготовки Z (см. <_AMODE> СЕДЬМАЯ ПОЗИЦИЯ)		
18	XA	<_XA>	REAL	Граница 1 X (абс., всегда диаметр)		
19	ZA	<_ZA>	REAL	Граница 1 Z (абс.)		
20	XB	<_XB>	REAL	Граница 2 X (см. <_AMODE> ВОСЬМАЯ ПОЗИЦИЯ)		
21	ZB	<_ZB>	REAL	Граница 2 Z (см. <_AMODE>ДЕВЯТАЯ ПОЗИЦИЯ)		
22	XDA	<_XDA>	REAL	Граница врезания 1 для 1-й позиции врезания на торцевой стороне (абс., всегда диаметр)		
23	XDB	<_XDB>	REAL	Граница врезания 2 для 1-й позиции врезания на торцевой стороне (абс., всегда диаметр)		
24	N	<_N>	INT	Количество выточек		
25	DP	<_DP>	REAL	Интервал выточек	Продольная выточка: параллельно оси Z	
					Поперечная выточка: параллельно оси X	
26	DI	<_DI>	REAL	Интервал для прерывания подачи	0 =	нет прерывания
					> 0 =	с прерыванием
27	sc	<_SC>	REAL	Безопасное расстояние для обхода препятствий, инкрементальное		
28	D2	<_DN>	INT	D-номер для 2-го резца, если не запрограммирован ⇒ D+1		

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение		
29		<_GMODE>	INT	Геометрический режим (нормирование запрограммированных гео-значений)		
				ПЕРВАЯ ПОЗИЦИЯ:	заре- зер- ви- ро- вано	
				ВТОРАЯ ПОЗИЦИЯ:	заре- зер- ви- ро- вано	
				ТРЕТЬЯ ПОЗИЦИЯ:	выбор обработки/только расчет стар- товой точки	
					0 =	обычная обработка (режима совместимости не требуется)
					1 =	обычная обработка
					2 =	расчет стартовой позиции - нет обработки (только для вызова из ShopMill/ShopTurn)
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	ограничение	
					0 =	нет
					1 =	да
				ПЯТАЯ ПОЗИЦИЯ:	вести границу 1 X	
					0 =	нет
					1 =	да
				ШЕСТАЯ ПОЗИЦИЯ:	вести границу 2 X	
					0 =	нет
					1 =	да
				СЕДЬМАЯ ПОЗИЦИЯ:	вести границу 1 Z	
					0 =	нет
					1 =	да
				ВОСЬМАЯ ПОЗИЦИЯ:	вести границу 2 Z	
					0 =	нет
					1 =	да

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
30		<_DMODE>	INT	Режим индикации
				ПЕРВАЯ ПОЗИЦИЯ:
				плоскость обработки G17/G18/G19
				0 = совместимость, остается активной действующая перед вызовом цикла плоскость
				1 = G17 (активна только в цикле)
				2 = G18 (активна только в цикле)
				3 = G19 (активна только в цикле)
				ВТОРАЯ ПОЗИЦИЯ:
				технологический режим
				1 = съём припуска вдоль контура
				2 = прорезание контура
				3 = токарная выточка
				ТРЕТЬЯ ПОЗИЦИЯ:
				обработка остаточного материала
				0 = нет
				1 = да
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:
				--- зарезервировано
				ПЯТАЯ ПОЗИЦИЯ:
				Технологическое масштабирование внутри масок цикла (Страница 870)
				0 = ввод: полный
				1 = ввод: простой

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение	
31		<_AMODE>	INT	Альтернативный режим	
				ПЕРВАЯ ПОЗИЦИЯ:	выбор подачи
					0 = подача DX и DZ при типе резания параллельно контуру
					1 = подача D
				ВТОРАЯ ПОЗИЦИЯ:	стратегия подачи
					0 = переменная глубина резания (90 ... 100 %)
					1 = постоянная глубина резания
				ТРЕТЬЯ ПОЗИЦИЯ:	подрез
					0 = равномерно
					1 = выравнивание по кромкам
				ЧЕТВЕРТАЯ ПОЗИЦИЯ:	Выбор припуска контура U1, двойной чистовой
					0 = нет
					1 = да
				ПЯТАЯ ПОЗИЦИЯ:	выбор обновления заготовки
					0 = нет
					1 = да
				ШЕСТАЯ ПОЗИЦИЯ:	выбор припуска к заготовке XD
					0 = абсолютная, значение поперечной оси в диаметре
					1 = инкрементальная, значение поперечной оси в радиусе
32	DCH	<_DCH>	REAL	СЕДЬМАЯ ПОЗИЦИЯ:	выбор припуска к заготовке ZD
					0 = абсолютная
					1 = инкрементальная
33	FS	<_FS>	REAL	ВОСЬМАЯ ПОЗИЦИЯ:	выбор границы 2 XB
					0 = абсолютная, значение поперечной оси в диаметре
					1 = инкрементальная, значение поперечной оси в радиусе
34				ДЕВЯТАЯ ПОЗИЦИЯ:	выбор границы 2 ZB
					0 = абсолютная
					1 = инкрементальная
35				ДЕСЯТАЯ ПОЗИЦИЯ:	
					0 = главный канал
					1 = добавочный канал
36				Номер смежного канала, если на станке есть более 2 каналов	
37				Сдвиг канала	
38				Подача чистовой обработки при комплексной обработке	

20.1.44 CYCLE4071 - Продольное шлифование с подачей в точке возврата

Синтаксис

CYCLE4071 (<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>, <S_A1>, <S_A2>)

Параметр

№	Параметр	Тип данных	Значение
1	<S_A>	REAL	Глубина подачи в начале
2	<S_B>	REAL	Глубина подачи в конце
3	<S_W>	REAL	Ширина шлифования
4	<S_U>	REAL	Время выхаживания
5	<S_I>	REAL	Подача на глубину
6	<S_K>	REAL	Подача для поперечной подачи
7	<S_H>	INT	Число повторов
8	<S_A1>	AXIS	Ось подачи (опция)или 1-я Гео-ось
9	<S_A2>	AXIS	Ось качания (опция) или 2-я Гео-ось

Функция

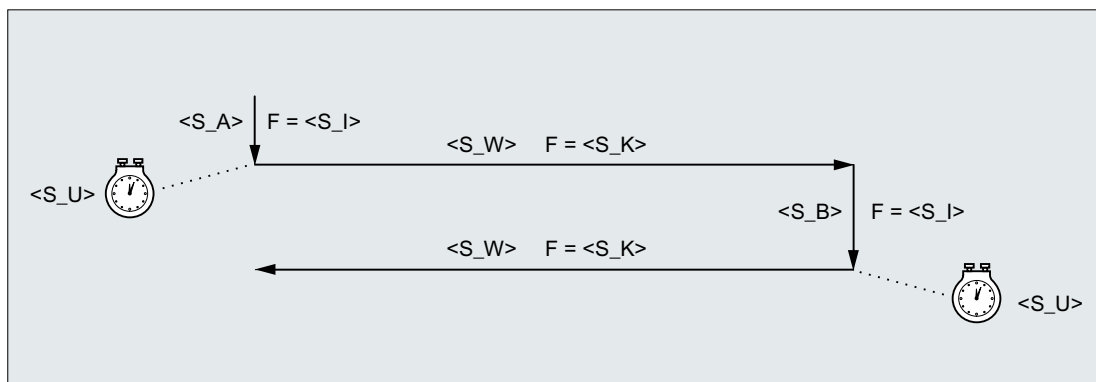
Цикл служит для обработки <S_H> подач. При этом глубина подачи в начале и в конце может быть разной. Во время подачи происходит движение по касательной.

Процесс

1. Начало цикла в текущей позиции оси качания.
2. Перемещение оси подачи на глубину подачи в начале P1 <S_A> с глубиной подачи P5 <S_I>.
3. Выхаживание со временем выхаживания P4 <S_U>.
4. Перемещение оси качания с шириной шлифования P3 <S_W> в качестве пути перемещения и подача для поперечной подачи P6 <S_K>.
5. Перемещение оси подачи на глубину подачи в конце P2 <S_B> с глубиной подачи P5 <S_I>.
6. Выхаживание со временем выхаживания P4 <S_U>.
7. Перемещение оси качания с шириной шлифования P4 <S_A> в качестве пути перемещения в начальную точку и подача для поперечной подачи P6 <S_K>.

Прервать этот процесс отдельным кадром невозможно.

Процесс повторяется в соответствии с запрограммированным количеством повторов P7 (<S_H>).



Пример

Выполнение двух маятниковых движений со следующими параметрами цикла:

- Начальная глубина подачи: 0,02 мм
- Конечная глубина подачи: 0,01 мм
- Ход: 100 мм
- Время выхаживания: 1 с
- Скорость подачи: 1 мм/мин
- Поперечная подача: 1000 мм/мин
- Повторы: 2
- Ось качания и ось подачи: стандартные гео-оси

Программный код

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.45 CYCLE4072 - Продольное шлифование с подачей в точке возврата и сигналом отмены

Синтаксис

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

Параметр

№	Параметр	Тип данных	Значение
1	<S_GAUGE>	STRING	Условия отмены подачи на глубину: 1. Номер быстрого входа 2. Логическое выражение
2	<S_A>	REAL	Глубина подачи в начале
3	<S_B>	REAL	Глубина подачи в конце
4	<S_W>	REAL	Ширина шлифования
5	<S_U>	REAL	Время выхаживания
6	<S_I>	REAL	Подача на глубину
7	<S_K>	REAL	Подача для поперечной подачи
8	<S_H>	INT	Число повторов
9	<S_A1>	AXIS	Ось подачи (опция)или 1-я Гео-ось
10	<S_A2>	AXIS	Ось качания (опция) или 2-я Гео-ось

Функция

Цикл служит для обработки <S_H> подач с учетом внешнего сигнала отмены. Глубина подачи в начале и в конце может различаться. Во время подачи происходит движение по касательной. Подача на глубину отменяется, если выполняется условие отмены. После отмены подачи на глубину всегда выполняется полный ход.

Процесс

1. Начало цикла в текущей позиции оси качания.
 2. Перемещение оси подачи на глубину подачи в начале P2 <S_A> с глубиной подачи P6 <S_I>.
 3. Выхаживание со временем выхаживания P5 <S_U>.
 4. Перемещение оси качания с шириной шлифования P4 <S_W> в качестве пути перемещения в начальную точку и подача для поперечной подачи P7 <S_K>.
 5. Перемещение оси подачи на глубину подачи в конце P3 <S_B> с глубиной подачи P6 <S_I>.
 6. Выхаживание со временем выхаживания P5 <S_U>.
 7. Перемещение оси качания с шириной шлифования P4 <S_W> в качестве пути перемещения в начальную точку и подача для поперечной подачи P7 <S_K>.
 8. Без отмены: Вышеуказанный процесс повторяется до тех пор, пока не будет достигнуто запрограммированное количество повторов P7 (<S_H>).
С отменой: Обработка завершается при достижении следующей исходной точки.
- Прервать этот процесс отдельным кадром невозможно.

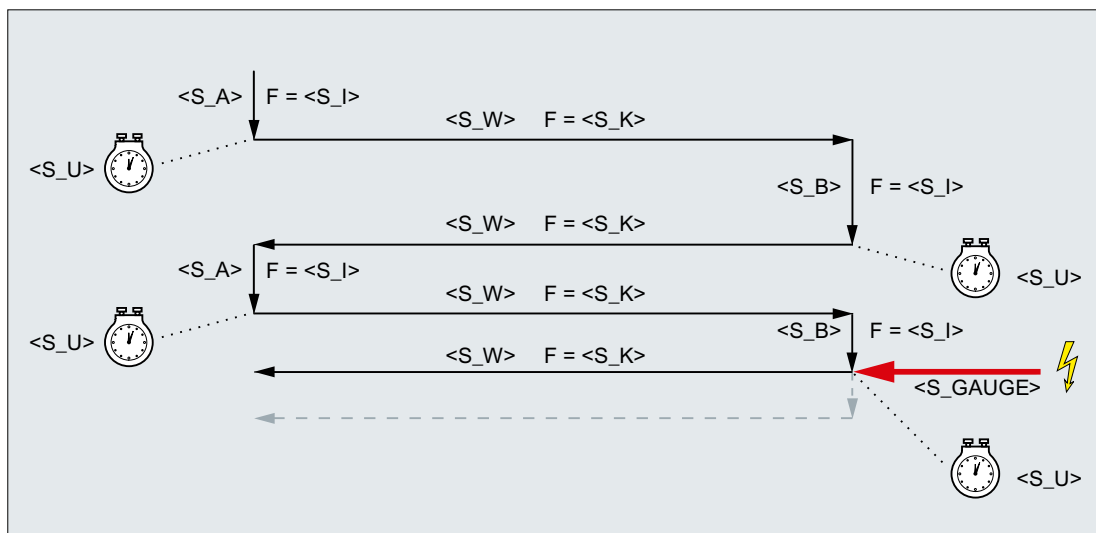


Рисунок 20-1 Отмена подачи на глубину в конце

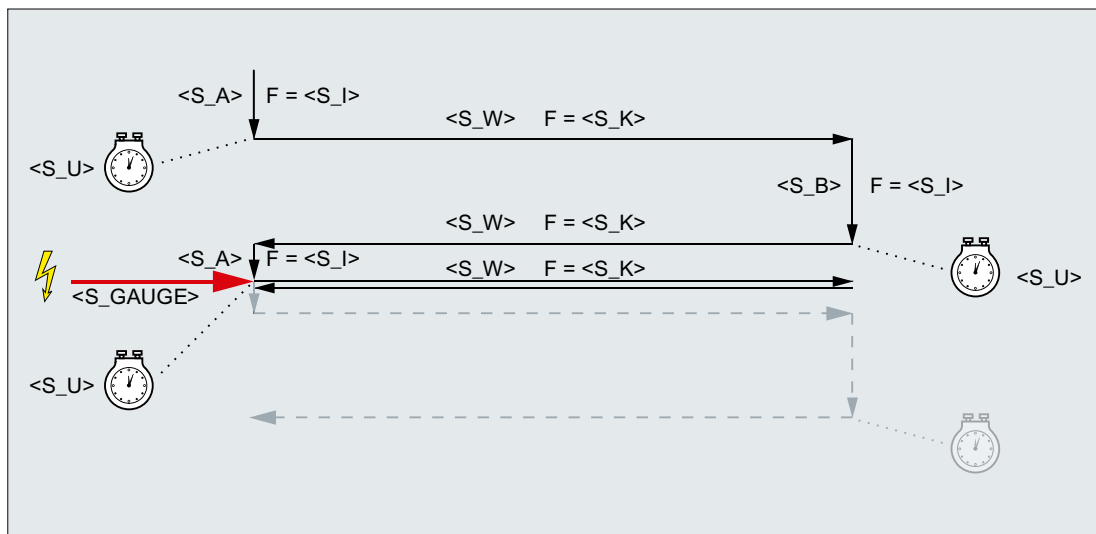


Рисунок 20-2 Отмена подачи на глубину в начале

Ресурсы

В качестве ресурсов цикл использует синхронное действие с выходом за границы кадров и переменную синхронного действия. Синхронное действие динамически определяется из свободной области диапазона синхронных действий (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR - 1199 ...). В качестве переменной синхронного действия используется SYG_IS[1].

Примеры

Пример 1: Осциллирующее движение с двумя ходами

Параметры цикла:

- Начальная глубина подачи: 0,02 мм
- Конечная глубина подачи: 0,01 мм
- Ход: 100 мм
- Время выхаживания: 1 с
- Скорость подачи: 1 мм/мин
- Поперечная подача: 1000 мм/мин
- Повторы: 2
- Ось качания и ось подачи: стандартные гео-оси

Сигнал отмены: быстрый вход 1 (\$A_IN[1])

Программный код

```
N10 T1 D1
N20 CYCLE4072 ("1", 0.02, 0.01, 100, 1, 1, 1000, 2)
N30 M30
```

Пример 2: Осциллирующее движение с двумя ходами

Параметры цикла:

- Начальная глубина подачи: 0,02 мм
- Конечная глубина подачи: 0,01 мм
- Ход: 100 мм
- Время выхаживания: 1 с
- Скорость подачи: 1 мм/мин
- Поперечная подача: 1000 мм/мин
- Повторы: 2
- Ось качания и ось подачи: стандартные гео-оси

Сигнал отмены: Переменная \$A_DBR[20] < 0,01

Программный код

```
N10 T1 D1
N20 CYCLE4072 ("($A_DBR[20]<0.01)", 0.02, 0.01, 100, 1, 1, 1000, 2)
N30 M30
```

20.1.46 CYCLE4073 - Продольное шлифование с непрерывной подачей на глубину**Синтаксис**

CYCLE4073 (<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>, <S_A2>)

Параметр

№	Параметр	Тип данных	Значение
1	<S_A>	REAL	Глубина подачи в начале
2	<S_B>	REAL	Глубина подачи в конце
3	<S_W>	REAL	Ширина шлифования
4	<S_U>	REAL	Время выхаживания
5	<S_K>	REAL	Подача для поперечной подачи
6	<S_H>	INT	Число повторов
7	<S_A1>	AXIS	Ось подачи (опция) или 1-я Гео-ось
8	<S_A2>	AXIS	Ось качания (опция) или 2-я Гео-ось

Функция

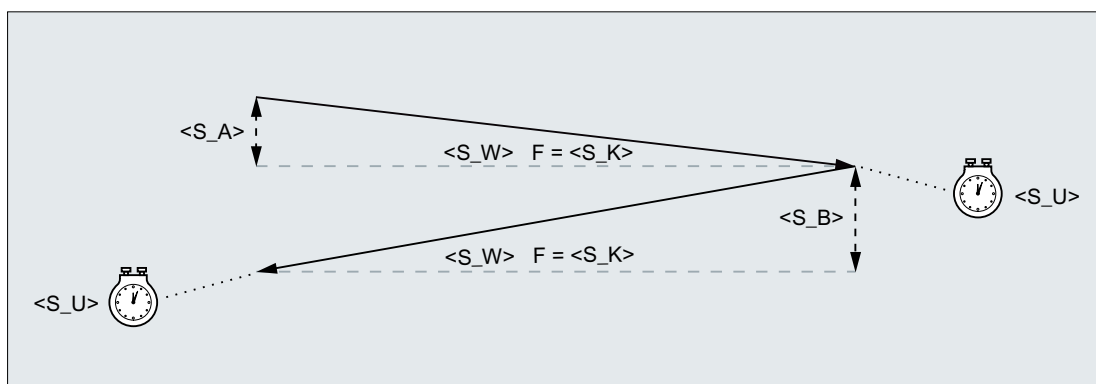
Цикл служит для выполнения <S_H> подач. При этом подача на глубину от начала к концу и от конца к началу может быть разной.

Процесс

1. Начало цикла в текущей позиции оси качания с глубиной подачи 0
2. Перемещение оси качания с шириной шлифования P3 <S_W> в качестве пути перемещения и подача для поперечной подачи P5 <S_K> с непрерывным увеличением глубины подачи до значения глубины подачи в начале P1 <S_A>.
3. Выхаживание со временем выхаживания P4 <S_U>.
4. Перемещение оси качания с шириной шлифования P3 <S_W> в качестве пути перемещения в начальную точку и подача для поперечной подачи P5 <S_K> с непрерывным увеличением глубины подачи до значения глубины подачи в конце P2 <S_B>.
5. Выхаживание со временем выхаживания P4 <S_U>.

Прервать этот процесс отдельным кадром невозможно.

Процесс повторяется в соответствии с запрограммированным количеством повторов P7 (<S_H>).



Пример

Осциллирующее движение с двумя ходами

Параметры цикла:

- Начальная глубина подачи: 0,02 мм
- Конечная глубина подачи: 0,01 мм
- Ход: 100 мм
- Время выхаживания: 1 с
- Поперечная подача: 1000 мм/мин
- Повторы: 2
- Ось качания и ось подачи: стандартные гео-оси

Программный код

```
N10 T1 D1
N20 CYCLE4073 (0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.47 CYCLE4074 - Продольное шлифование с непрерывной подачей на глубину и сигналом отмены

Синтаксис

```
CYCLE4074 (<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

Параметр

№	Параметр	Тип данных	Значение
1	<S_GAUGE>	STRING	Условия отмены подачи на глубину: 1. Номер быстрого входа 2. Логическое выражение
2	<S_A>	REAL	Глубина подачи в начале
3	<S_B>	REAL	Глубина подачи в конце
4	<S_W>	REAL	Ширина шлифования
5	<S_U>	REAL	Время выхаживания
6	<S_K>	REAL	Подача для поперечной подачи
7	<S_H>	INT	Число повторов
8	<S_A1>	AXIS	Ось подачи (опция) или 1-я Гео-ось
9	<S_A2>	AXIS	Ось качания (опция) или 2-я Гео-ось

Функция

Цикл служит для обработки <S_H> подач с учетом, например, внешнего сигнала отмены. Глубина подачи в начале и в конце может различаться. Подача на глубину отменяется, если выполняется условие отмены. После отмены подачи на глубину всегда выполняется полный ход.

Процесс

1. Начало цикла в текущей позиции оси качания с глубиной подачи 0
2. Перемещение оси качания с шириной шлифования P4 <S_W> в качестве пути перемещения и подача для поперечной подачи P6 <S_K> с непрерывным увеличением глубины подачи до значения глубины подачи в начале P2 <S_A>.
3. Выхаживание со временем выхаживания P5 <S_U>.
4. Перемещение оси качания с шириной шлифования P4 <S_W> в качестве пути перемещения в начальную точку и подача для поперечной подачи P6 <S_K> с непрерывным увеличением глубины подачи до значения глубины подачи в конце P3 <S_B>.
5. Выхаживание со временем выхаживания P5 <S_U>.
6. Без отмены: Вышеуказанный процесс повторяется до тех пор, пока не будет достигнуто запрограммированное количество повторов P7 (<S_H>).
С отменой: Подача на глубину прерывается. Обработка завершается при достижении следующей исходной точки.

Прервать этот процесс отдельным кадром невозможно.

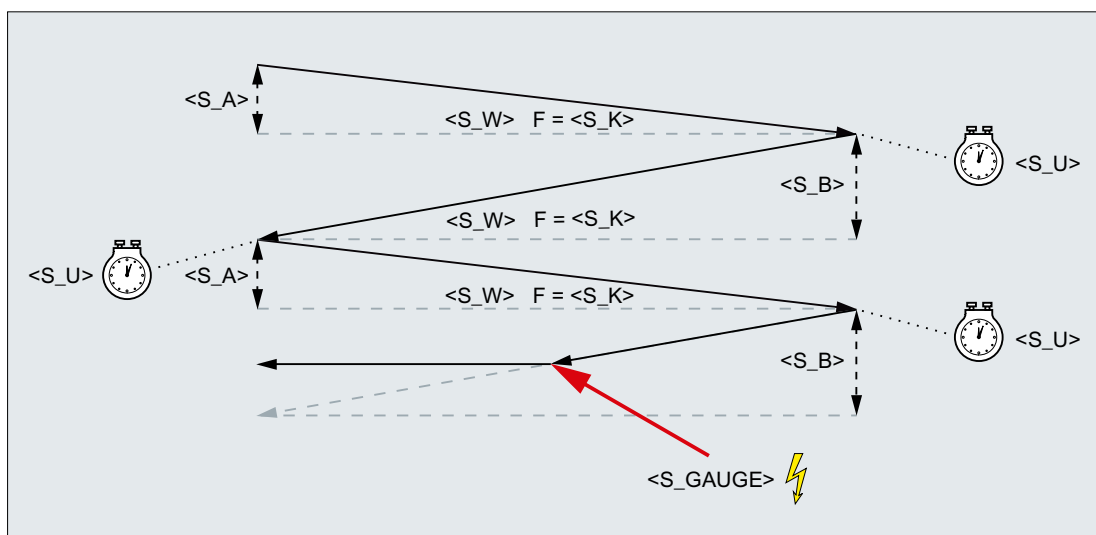


Рисунок 20-3 Отмена подачи на глубину от конца к началу

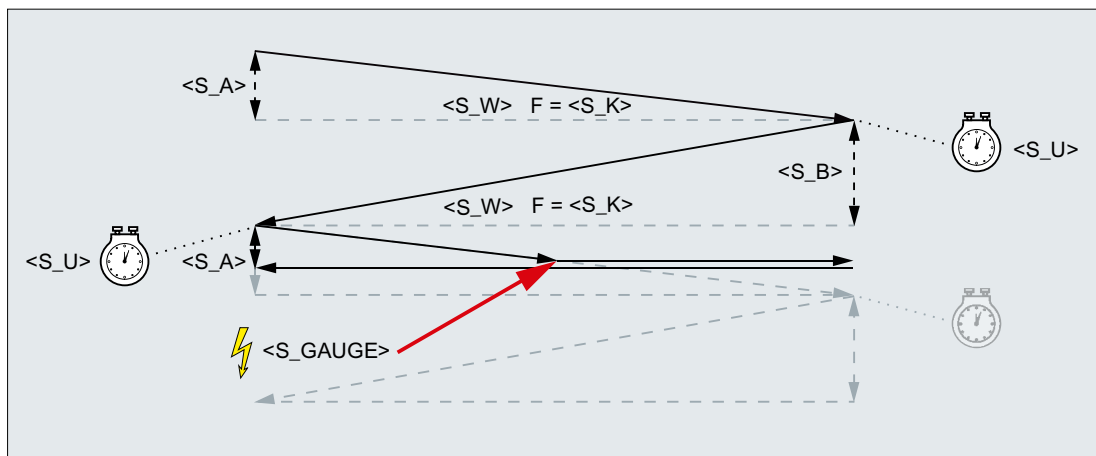


Рисунок 20-4 Отмена подачи на глубину от начала к концу

Ресурсы

В качестве ресурсов цикл использует синхронное действие с выходом за границы кадров и переменную синхронного действия. Синхронное действие динамически определяется из свободной области диапазона синхронных действий (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR - 1199 ...). В качестве переменной синхронного действия используется SYG_IS[1].

Примеры

Пример 1: Осциллирующее движение с двумя ходами

Параметры цикла:

- Начальная глубина подачи: 0,02 мм
- Конечная глубина подачи: 0,01 мм

- Ход: 100 мм
- Время выхаживания: 1 с
- Поперечная подача: 1000 мм/мин
- Повторы: 2
- Ось качания и ось подачи: стандартные гео-оси

Сигнал отмены: быстрый вход 1 (\$A_IN[1])

Программный код

```
N10 T1 D1
N20 CYCLE4074 ("1", 0.02, 0.01, 100, 1, 1000, 2)
N30 M30
```

Пример 2: Осциллирующее движение с двумя ходами

Параметры цикла:

- Начальная глубина подачи: 0,02 мм
- Конечная глубина подачи: 0,01 мм
- Ход: 100 мм
- Время выхаживания: 1 с
- Поперечная подача: 1000 мм/мин
- Повторы: 2
- Ось качания и ось подачи: стандартные гео-оси

Сигнал отмены: Переменная \$A_DBR[20] < 0,01

Программный код

```
N10 T1 D1
N20 CYCLE4074 ("($A_DBR[20]<0.01)", 0.02, 0.01, 100, 1, 1000, 2)
N30 M30
```

20.1.48 CYCLE4075 - Плоское шлифование с подачей в точке возврата

Синтаксис

```
CYCLE4075 (<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Параметр

№	Параметр	Тип данных	Значение
1	<S_I>	REAL	Глубина подачи в начале
2	<S_J>	REAL	Глубина подачи в конце

№	Параметр	Тип данных	Значение
3	<S_K>	REAL	Общая глубина подачи
4	<S_A>	REAL	Ширина шлифования
5	<S_R>	REAL	Подача на глубину
6	<S_F>	REAL	Подача для поперечной подачи
7	<S_P>	REAL	Время выхаживания
8	<S_A1>	AXIS	Ось подачи (опция)
9	<S_A2>	AXIS	Качающаяся ось (опция)

Функция

Цикл служит для выполнения обработки с общей глубиной подачи P3 <S_K> в несколько этапов подачи. Глубина подачи в начале P1 <S_I> и конце P2 <S_J> может различаться. Во время подачи происходит движение по касательной.

Параметры пути P1 - P4 могут иметь отрицательное или положительное значение.

Данные оси подачи P8 <S_A1> и / или оси качания P9 <S_A2> не обязательны. Если один или оба параметра не заданы, то цикл использует обе первые гео-оси канала.

Если сумма глубины подачи в начале P1 <S_I> и в конце P2 <S_J> равна 0, или общая глубина подачи P3 <S_K> равна 0, то выполняется только один ход выхаживания.

Процесс

1. Начало цикла в текущей позиции оси качания.
2. Перемещение оси подачи на глубину подачи в начале P1 <S_I> с глубиной подачи P5 <S_R>.
3. Выхаживание со временем выхаживания P7 <S_P>
4. Перемещение оси качания с шириной шлифования P4 <S_A> в качестве пути перемещения в начальную точку и подача для поперечной подачи P6 <S_F>.
5. Перемещение оси подачи на глубину подачи в конце P2 <S_J> с глубиной подачи P5 <S_R>.
6. Выхаживание со временем выхаживания P7 <S_P>.
7. Перемещение оси качания с шириной шлифования P4 <S_A> в качестве пути перемещения в начальную точку и подача для поперечной подачи P6 <S_F>.

Прервать этот процесс отдельным кадром невозможно.

Процесс повторяется до тех пор, пока не будет достигнута общая глубина подачи P3 <S_K>. Последний ход в этом случае разделяется неравномерно.

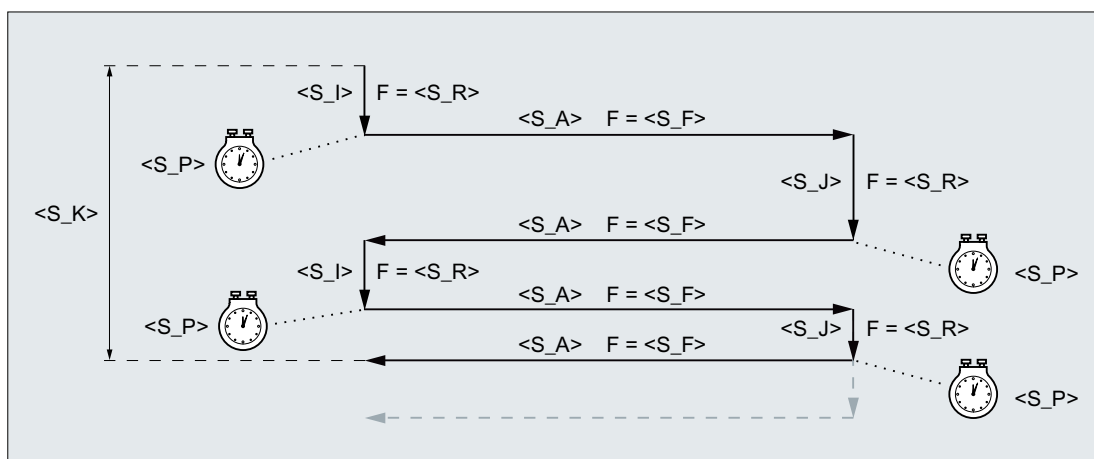


Рисунок 20-5 Общая глубина подачи достигается при подаче на второй точке возврата

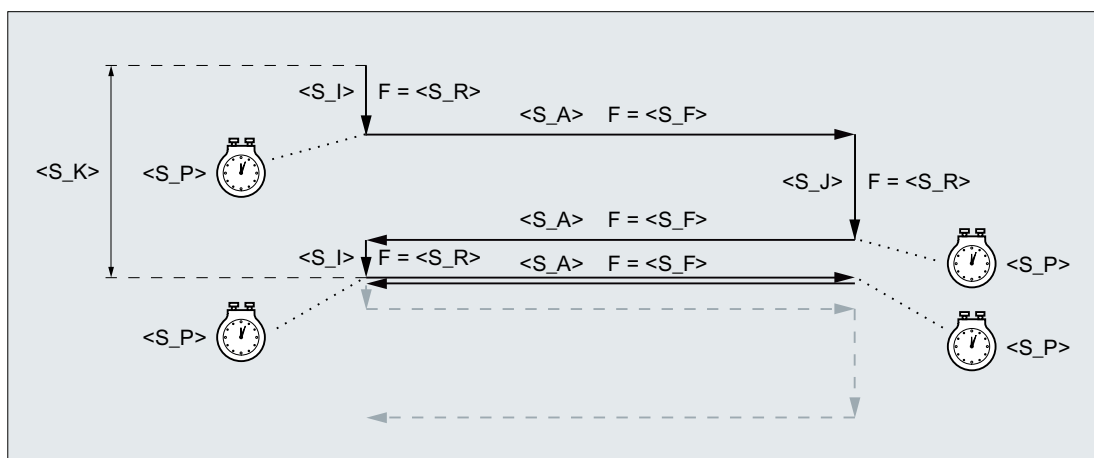


Рисунок 20-6 Общая глубина подачи достигается при подаче на первой точке возврата

Пример

Осциллирующее движение со следующими данными:

- глубина подачи 0,02 мм в начале
- глубина подачи 0,01 мм в конце
- общая глубина подачи 1 мм
- ход 100 мм
- подача на глубину 1 мм/мин
- поперечная подача 1000 мм/мин
- время выхода 1 с
- стандартные гео-оси

Программный код

```
N10 T1 D1
```

Программный код

```
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.49 CYCLE4077 - Плоское шлифование с подачей в точке возврата и сигналом отмены

Синтаксис

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

Параметр

№	Параметр	Тип данных	Значение
1	<S_GAUGE>	STRING	Условие отмены подачи на глубину: • Номер быстрого входа • Логическое выражение
2	<S_I>	REAL	Глубина подачи в начале
3	<S_J>	REAL	Глубина подачи в конце
4	<S_K>	REAL	Общая глубина подачи
5	<S_A>	REAL	Ширина шлифования
6	<S_R>	REAL	Подача на глубину
7	<S_F>	REAL	Подача для поперечной подачи
8	<S_P>	REAL	Время выхаживания
9	<S_A1>	AXIS	Ось подачи (опция)
10	<S_A2>	AXIS	Качающаяся ось (опция)

Функция

Цикл служит для выполнения обработки с общей глубиной подачи P4 <S_K> в несколько этапов подачи. Глубина подача в начале P2 <S_I> и конце P3 <S_J> может различаться. Во время подачи происходит движение по касательной. Подача на глубину прекращается, когда сигнал отмены быстрого входа равен 1, или выполняется условие отмены. После отмены выполняется полный ход.

Параметры пути P2 - P5 могут иметь отрицательное или положительное значение.

Данные оси подачи P9 <S_A1> и / или оси качания P10 <S_A2> не обязательны. Если один или оба параметра не заданы, то цикл использует обе первые гео-оси канала.

Если сумма глубины подачи в начале P2 <S_I> и в конце P3 <S_J> равна 0, или общая глубина подачи P4 <S_K> равна 0, то выполняется только один ход выхаживания.

Процесс

1. Начало цикла в текущей позиции оси качания.
2. Перемещение оси подачи на глубину подачи в начале P2 <S_I> с глубиной подачи P6 <S_R>.
3. Выхаживание со временем выхаживания P8 <S_P>.
4. Перемещение оси качания с шириной шлифования P5 <S_A> в качестве пути перемещения в начальную точку и подача для поперечной подачи P7 <S_F>.
5. Перемещение оси подачи на глубину подачи в конце P3 <S_J> с глубиной подачи P6 <S_R>.
6. Выхаживание со временем выхаживания P8 <S_P>.
7. Перемещение оси качания с шириной шлифования P5 <S_A> в качестве пути перемещения в начальную точку и подача для поперечной подачи P7 <S_F>.
8. Без отмены: Вышеописанный процесс повторяется до тех пор, пока не будет достигнута общая глубина подачи P4 <S_K>. Последний ход в этом случае разделяется неравномерно.
С отменой: Обработка завершается в исходной точке.

Прервать этот процесс отдельным кадром невозможно.

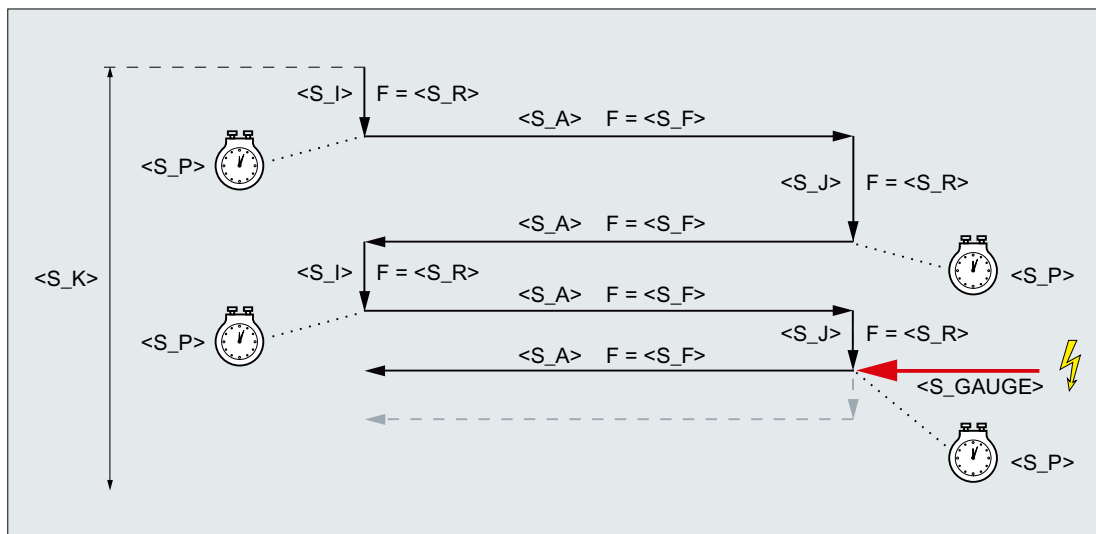


Рисунок 20-7 Отмена подачи на глубину в конце

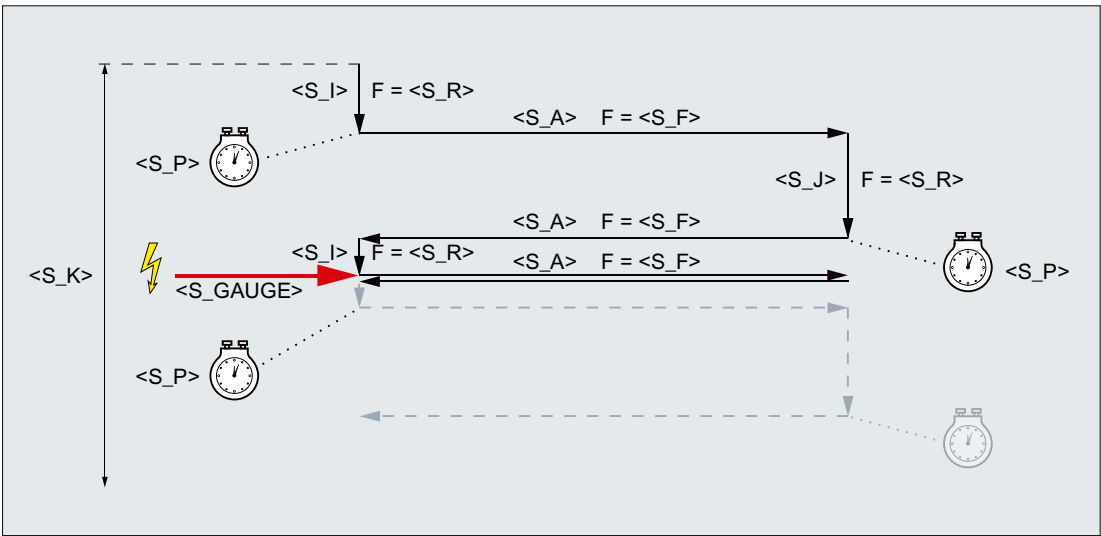


Рисунок 20-8 Отмена подачи на глубину в начале

Ресурсы

В качестве ресурсов цикл использует синхронное действие с выходом за границы кадров и переменную синхронного действия. Синхронное действие динамически определяется из свободной области диапазона синхронных действий (CUS.DIR - 1 ..., CMA.DIR - 1000 ... , CST.DIR - 1199 ...). В качестве переменной синхронного действия используется SYG_IS[1].

Примеры

Пример 1

Осциллирующее движение со следующими данными:

- глубина подачи 0,02 мм в начале
- глубина подачи 0,01 мм в конце
- общая глубина подачи 1 мм
- ход 100 мм
- подача на глубину 1 мм/мин
- поперечная подача 1000 мм/мин
- время выхаживания 1 с
- стандартные гео-оси

Сигнал отмены: быстрый вход 1 (\$A_IN[1])

Программный код

```
N10 T1 D1
N20 CYCLE4077 ("1", 0.02, 0.01, 1, 100, 1, 1000, 1)
N30 M30
```

Пример 2

Осциллирующее движение со следующими данными:

- глубина подачи 0,02 мм в начале
- глубина подачи 0,01 мм в конце
- общая глубина подачи 1 мм
- ход 100 мм
- подача на глубину 1 мм/мин
- поперечная подача 1000 мм/мин
- время выхаживания 1 с
- стандартные гео-оси

Сигнал отмены: Dualport-RAM-переменная 20 меньше, чем 0,01 (\$A_DBR[20] < 0,01)

Программный код

```
N10 T1 D1
N20 CYCLE4077("($A_DBR[20]<0.01)",0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.50 CYCLE4078 - Плоское шлифование с непрерывной подачей на глубину**Синтаксис**

CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>, <S_A2>)

Параметр

№	Параметр	Тип данных	Значение
1	<S_I>	REAL	Глубина подачи от начала к концу
2	<S_J>	REAL	Глубина подачи от конца к началу
3	<S_K>	REAL	Общая глубина подачи
4	<S_A>	REAL	Ширина шлифования
5	<S_F>	REAL	Подача
6	<S_P>	REAL	Время выхаживания
7	<S_A1>	AXIS	Ось подачи (опция)
8	<S_A2>	AXIS	Качающаяся ось (опция)

Функция

Цикл служит для выполнения обработки с общей глубиной подачи P3 <S_K> путем непрерывной подачи. Глубина подача от начала к концу P1 <S_I> и от конца к началу P2 <S_J> может различаться.

Параметры пути P1 - P4 могут иметь отрицательное или положительное значение.

Данные оси подачи P8 <S_A1> и / или оси качания P9 <S_A2> не обязательны. Если один или оба параметра не заданы, то цикл использует обе первые гео-оси канала.

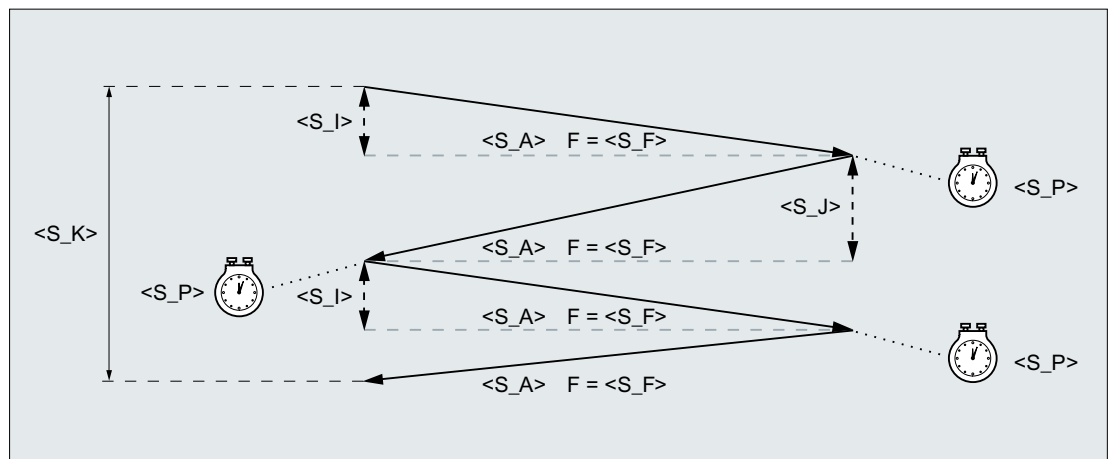
Если сумма глубины подачи P1 <S_I> и P2 <S_J> равна 0, или общая глубина подачи P3 <S_K> равна 0, то выполняется только один ход выхаживания.

Процесс

1. Начало цикла в текущей позиции оси качания с глубиной подачи 0
2. Перемещение оси качания с шириной шлифования P4 <S_A> в качестве пути перемещения и подача P5 <S_F> с непрерывным увеличением глубины подачи до значения глубины подачи в начале P1 <S_I>.
3. Выхаживание со временем выхаживания P7 <S_P>
4. Перемещение оси качания с шириной шлифования P4 <S_A> в качестве пути перемещения в начальную точку и подача P5 <S_F> с непрерывным увеличением глубины подачи до значения глубины подачи в конце P2 <S_J>.
5. Выхаживание со временем выхаживания P7 <S_P>.
6. Перемещение оси качания с шириной шлифования P4 <S_A> в качестве пути перемещения в начальную точку и подача P5 <S_F>.

Прервать этот процесс отдельным кадром невозможно.

Процесс повторяется до тех пор, пока не будет достигнута общая глубина подачи P3 <S_K>. Последний ход в этом случае разделяется неравномерно.



Пример

Осциллирующее движение со следующими данными:

- глубина подачи 20 мм в начале
- глубина подачи 10 мм в конце
- общая глубина подачи 100 мм
- ход 100 мм

- Подача 1000 мм/мин
- время выхаживания 1 с
- стандартные гео-оси

Программный код

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

20.1.51 CYCLE4079 - Плоское шлифование с прерывистой подачей на глубину

Синтаксис

```
CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>, <S_A2>)
```

Параметр

№	Параметр	Тип данных	Значение
1	<S_I>	REAL	Глубина подачи в начале
2	<S_J>	REAL	Глубина подачи в конце
3	<S_K>	REAL	Общая глубина подачи
4	<S_A>	REAL	Ширина шлифования
5	<S_R>	REAL	Подача на глубину
6	<S_F>	REAL	Подача для поперечной подачи
7	<S_P>	REAL	Время выхаживания
8	<S_A1>	AXIS	Ось подачи (опция)
9	<S_A2>	AXIS	Качающаяся ось (опция)

Функция

Цикл служит для выполнения обработки с общей глубиной подачи P3 <S_K> в несколько этапов подачи. Глубина подачи в начале P1 <S_I> и конце P2 <S_J> может различаться. Во время подачи происходит движение по касательной.

Параметры пути P1 - P4 могут иметь отрицательное или положительное значение.

Данные оси подачи P8 <S_A1> и / или оси качания P9 <S_A2> не обязательны. Если один или оба параметра не заданы, то цикл использует обе первые гео-оси канала.

Если сумма глубины подачи в начале P1 <S_I> и в конце P2 <S_J> равна 0, или общая глубина подачи P3 <S_K> равна 0, то выполняется только один ход выхаживания.

Процесс

1. Начало цикла в текущей позиции оси качания.
2. Перемещение оси подачи на глубину подачи в начале P1 <S_I> с глубиной подачи P5 <S_R>.
3. Выхаживание со временем выхаживания P7 <S_P>
4. Перемещение оси качания с шириной шлифования P4 <S_A> в качестве пути перемещения в начальную точку и подача для поперечной подачи P6 <S_F>.
5. Перемещение оси подачи на глубину подачи в конце P2 <S_J> с глубиной подачи P5 <S_R>.
6. Выхаживание со временем выхаживания P7 <S_P>.
7. Перемещение оси качания с шириной шлифования P4 <S_A> в качестве пути перемещения в начальную точку и подача для поперечной подачи P6 <S_F>.

Прервать этот процесс отдельным кадром невозможно.

Процесс повторяется до тех пор, пока не будет достигнута общая глубина подачи P3 <S_K>. Последний ход в этом случае разделяется неравномерно.

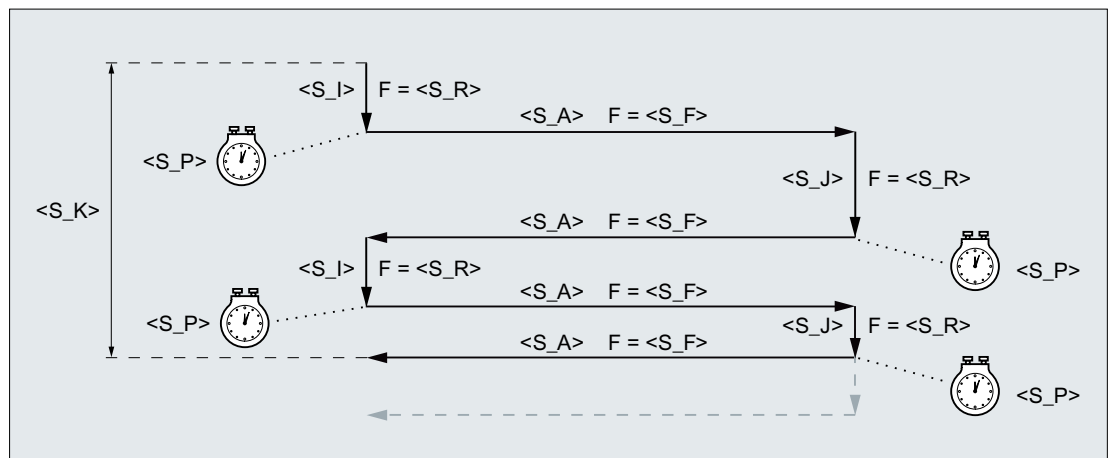


Рисунок 20-9 Общая глубина подачи достигается при подаче на второй точке возврата

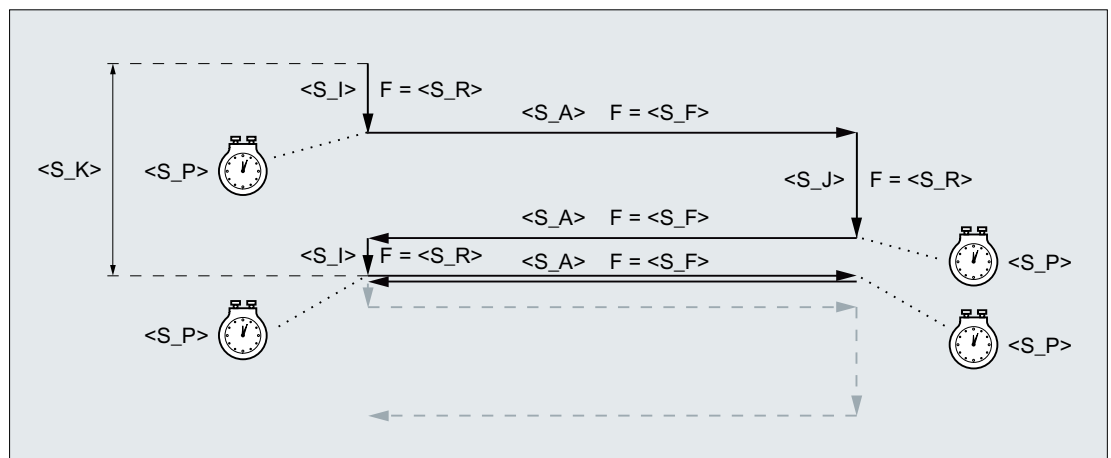


Рисунок 20-10 Общая глубина подачи достигается при подаче на первой точке возврата

Пример

Осциллирующее движение со следующими данными:

- глубина подачи 0,02 мм в начале
- глубина подачи 0,01 мм в конце
- общая глубина подачи 1 мм
- ход 100 мм
- подача на глубину 1 мм/мин
- поперечная подача 1000 мм/мин
- время выхаживания 1 с
- стандартные гео-оси

Программный код

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.52 GROUP_BEGIN — Начало блока программы

Синтаксис

GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
1		<_LEVEL>	INT	Плоскость
				0 = Главная плоскость
				1 = Вспомогательный уровень 1
2		<_NAME>	STRING[128]	Имя блока
3		<_SP>	INT	Шпиндель
				0 = Шпинделя нет
				1 = Главный шпиндель
				2 = Противושпиндель
4		<_MODE>	INT	Режим
				Бит 0 =1 GROUP_ADDEND существует
				Бит 1 =1 ShopTurn: автоматический отвод (движение к точке смены инструмента)
				Бит 12 зарезервировано
				Бит 13 зарезервировано
5		<S_ICON>	STRING[32]	Название иконки (только для интерфейса пользователя)

20.1.53 GROUP_END — Конец блока программы

Синтаксис

```
GROUP_END(<_LEVEL>, <_SP>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
1		<_LEVEL>	INT	Плоскость
				0 = Главная плоскость
				1 = Вспомогательный уровень 1
2		<_SP>	INT	Шпиндель
				0 = Шпинделя нет
				1 = Главный шпиндель
				2 = Противопиндель

20.1.54 GROUP_ADDEND — конец дополнения ввода

Синтаксис

```
GROUP_ADDEND(<_LEVEL>, <_SP>)
```

Параметр

№	Параметр в маске	Параметр внутренний	Тип данных	Объяснение
1		<_LEVEL>	INT	Плоскость
				0 = Главная плоскость
				1 = Вспомогательный уровень 1
2		<_SP>	INT	Шпиндель
				0 = Шпинделя нет
				1 = Главный шпиндель
				2 = Противопиндель

20.1.55 Граничные условия

20.1.55.1 Технологическое масштабирование внутри масок цикла

Если технологическое масштабирование активно, для различных масок цикла можно выбрать упрощенный ввод, при котором отображаются только самые важные параметры цикла.

Упрощенный ввод возможен, например, в следующих масках цикла:

Технология	Маска цикла
Сверление	Глубокое сверление
	Нарезание внутренней резьбы
Фрезерная обработка	Прямоугольный карман
	Контурное фрезерование: карман
Токарная обработка	Нарезание резьбы резцом: продольное
	Токарная обработка контура: обработка резанием
	Токарная обработка контура: выточка
	Токарная обработка контура: токарная выточка

Для соответствующих масок цикла в интерфейсе пользователя есть опции "Ввод: простой" и "Ввод: полный".

Неотображаемые параметры цикла

Либо не отображаемые при упрощенном вводе параметры цикла предустанавливаются с фиксированными, технологически целесообразными, но не изменяемыми значениями. Либо параметрам цикла через специфические для канала установочные данные цикла присваиваются параметрируемые значения. См. ниже абзац "Ввод в эксплуатацию" > "Спец. для канала установочные данные циклов"

Переключение "Ввод: полный" > "Ввод: простой"

Если одна из масок цикла заполняется при настройке "Ввод полный", а затем переключается на "Ввод простой", то при генерировании вызова цикла для более не отображаемых параметров используются значения по умолчанию или значения установочных данных.

Ввод в эксплуатацию

Спец. для канала машинные данные конфигурации

С помощью машинных данных активируется возможность технологического масштабирования внутри масок цикла:

MD52210 \$MCS_FUNCTION_MASK_DISP, бит 9 = 1 (отображать выбор "Ввод простой")

Спец. для канала установочные данные циклов

Если упрощенный ввод внутри масок цикла активен, значения для определенных параметров цикла можно задавать через следующие установочные данные:

Номер	Идентификатор	Объяснение
SD55300	\$SCS_EASY_SAFETY_CLEARANCE	Безопасное расстояние
SD55301	\$SCS_EASY_DWELL_TIME	Время ожидания
SD55305	\$SCS_EASY_DRILL_DEEP_FD1	Глубокое сверление: процент: 1-я подача
SD55306	\$SCS_EASY_DRILL_DEEP_DF	Глубокое сверление: процент: подача
SD55307	\$SCS_EASY_DRILL_DEEP_V1	Глубокое сверление: мин. подача на глубину
SD55308	\$SCS_EASY_DRILL_DEEP_V2	Глубокое сверление: величина отвода
SD55309	\$SCS_EASY_THREAD_RETURN_DIST	Нарезание резьбы резцом: интервал обратного хода

20.2 Измерительные циклы

Измерительные циклы – это специальные подпрограммы Siemens для решения различных задач измерения. Как и любые другие циклы, измерительные циклы тоже можно адаптировать к конкретной проблеме с помощью параметров.

Измерительные циклы можно использовать для измерений в следующих областях и технологиях:

- измерение инструментов для токарной обработки / фрезерования
- измерение деталей при токарной обработке / фрезеровании

Литература

Подробное описание измерительных циклов см.:

Руководство по программированию "Измерительные циклы"

Таблицы

21.1 Операторы

Примечание

Циклы

В списке операторов содержатся все циклы, которые могут появиться в программе ЧПУ (G-код), т. е. которые можно программировать в редакторе программ через маски или которые требуется программировать при шлифовании без программной поддержки. Не учтены циклы, все еще представленные в системе управления из соображений совместимости, но которые больше не поддаются программированию в редакторе программ SINUMERIK Operate ("Циклы совместимости").

Операторы A ... C

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
:	O	Номер главного кадра ЧПУ, конец метки перехода, связывающий оператор		+		PGAsl
*	O	Оператор для умножения		+		PGAsl
+	O	Оператор для сложения		+		PGAsl
-	O	Оператор для вычитания		+		PGAsl
<	O	Оператор сравнения, меньше		+		PGAsl
<<	O	Связывающий оператор для строк		+		PGAsl
<=	O	Оператор сравнения, меньше равно		+		PGAsl
=	O	Оператор присваивания		+		PGAsl
>=	O	Оператор сравнения, больше равно		+		PGAsl
/	O	Оператор для деления		+		PGAsl
/0		Кадр пропускается (1-й уровень пропуска)°		+		PGsl
...		...				
...		...				
/7		Кадр пропускается (8-й уровень пропуска)				
A	A	Имя оси	м/п	+		PGAsl
A2	A	Ориентация инструмента: Угол RPY или эйлеров угол	п	+		PGAsl
A3	A	Ориентация инструмента: 1-я компонента вектора направления	п	+		PGAsl
A4	A	Ориентация инструмента: 1-я компонента вектора нормали к плоскости на начале кадра	п	+		PGAsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
A5	A	Ориентация инструмента: 1-я компонента вектора нормали к плоскости на конце кадра	п	+		PGAsI
A6	A	Ориентация инструмента: 1-я компонента вектора направления для оси вращения конуса	п	+		PGAsI
A7	A	Ориентация инструмента: 1-я компонента вектора для промежуточной ориентации на боковой поверхности конуса	п	+		PGAsI
ABS	F	Абсолютное значение (величина)		+	+	PGAsI
AC	K	Абсолютное указание размеров координат/позиций	п	+		PGsI
ACC	K	Управление актуальным осевым ускорением	м	+	+	PGsI
ACCLIMA	K	Управление актуальным максимальным осевым ускорением	м	+	+	PGAsI
ACN	K	Абсолютное указание размера для круговых осей, подвод к позиции в отрицательном направлении	п	+		PGsI
ACOS	F	Арккосинус (тригонометрическая функция)		+	+	PGAsI
ACP	K	Абсолютное указание размера для круговых осей, подвод к позиции в положительном направлении	п	+		PGsI
ACTBLOCNO	P	Вывод актуального номера кадра ошибки, даже в случае активности "актуальная индикация кадра подавлена" (DISPLOF)!		+		PGAsI
ADDFRAME	F	Учет и возможная активация измеренного фрейма		+	-	PGAsI, FB1sI (K2)
ADIS	A	Интервал перешлифовки для траекторных функций G1, G2, G3, ...	м	+		PGsI
ADISPOS	A	Интервал перешлифовки для ускоренного хода G0	м	+		PGsI
ADISPOSA	P	Размер окна допуска для IPOBRKA	м	+	+	PGAsI
ALF	A	Угол быстрого отвода	м	+		PGAsI
AMIRROR	G	Программируемое отражение	п	+		PGsI
AND	K	Логическая И		+		PGAsI
ANG	A	Угол линии контура	п	+		PGsI
AP	A	Полярный угол	м/п	+		PGsI
APR	K	Чтение / индикация защиты доступа		+		PGAsI
APRB	K	Чтение права доступа, BTSS		+		PGAsI
APRP	K	Чтение права доступа, программа обработки детали		+		PGAsI
APW	K	Запись защиты доступа		+		PGAsI
APWB	K	Запись права доступа, BTSS		+		PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
APWP	K	Запись права доступа, программа обработки детали		+		PGAsI
APX	K	Определение защиты доступа для выполнения указанного элемента языка		+		PGAsI
AR	A	Апертурный угол	м/п	+		PGsI
AROT	G	Программируемое вращение	п	+		PGsI
AROTS	G	Программируемые вращения фреймов с пространственными углами	п	+		PGsI
AS	K	Макро-определение		+		PGAsI
ASCALE	G	Программируемое масштабирование	п	+		PGsI
ASIN	F	Математическая функция, арксинус		+	+	PGAsI
ASPLINE	G	Акима-сплайн	м	+		PGAsI
ATAN2	F	Арктангенс2		+	+	PGAsI
ATOL	A	Спец. для оси допуск для функций компрессора, сглаживания ориентации и типов перешлифовки	м	+		PGAsI
ATRANS	G	Аддитивное программируемое смещение нулевой точки	п	+		PGsI
AUXFUDEL	P	Удалить вспомогательную функцию спец. для канала из глобального списка		+	-	FB1sI (H2)
AUXFUDELG	P	Удалить все вспомогательные функции группы вспомогательных функций спец. для канала из глобального списка		+	-	FB1sI (H2)
AUXFUMSEQ	P	Определить последовательность вывода для вспомогательных функций M		+	-	FB1sI (H2)
AUXFUSYNC	P	Создать из глобального списка вспомогательных функций полный кадр программы обработки детали для спец. для канала SERUPRO-Ende-ASUP		+	-	FB1sI (H2)
AX	K	Переменный идентификатор оси	м/п	+		PGAsI
AXCTSWE	P	Вращать осевой контейнер		+	-	PGAsI
AXCTSWEC	P	Отмена разрешения на вращение осевого контейнера		+	+	PGAsI
AXCTSWED	P	Вращать осевой контейнер (командная переменная для ввода в эксплуатацию!)		+	-	PGAsI
AXIS	K	Идентификатор оси, адрес оси		+		PGAsI
AXNAME	F	Преобразует входную строку в идентификатор оси		+	-	PGAsI
AXSTRING	F	Преобразует строку номер шпинделя		+	-	PGAsI
AXTOCHAN	P	Затребовать ось для определенного канала. Возможно из программы ЧПУ и из синхронного действия.		+	+	PGAsI
AXTOSPI	F	Преобразует идентификатор оси в индекс шпинделя		+	-	PGAsI
B	A	Имя оси	м/п	+		PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
B2	A	Ориентация инструмента: Угол RPY или эйлеров угол	п	+		PGAsI
B3	A	Ориентация инструмента: Компонента вектора направления	п	+		PGAsI
B4	A	Ориентация инструмента: 2-я компонента вектора нормали к плоскости на начале кадра	п	+		PGAsI
B5	A	Ориентация инструмента: 2-я компонента вектора нормали к плоскости на конце кадра	п	+		PGAsI
B6	A	Ориентация инструмента: 2-я компонента вектора направления для оси вращения конуса	п	+		PGAsI
B7	A	Ориентация инструмента: 2-я компонента вектора для промежуточной ориентации на боковой поверхности конуса	п	+		PGAsI
B_AND	O	Побитовая И		+		PGAsI
B_OR	O	Побитовая ИЛИ		+		PGAsI
B_NOT	O	Побитовое отрицание		+		PGAsI
B_XOR	O	Побитовая исключающая "ИЛИ"		+		PGAsI
BAUTO	G	Определение первого сегмента сплайна через следующие 3 точки	м	+		PGAsI
BLOCK	K	Определяет вместе с кодовым словом ТО обрабатываемый программный блок в косвенном вызове подпрограммы		+		PGAsI
BLSYNC	K	Выполнение обработчика прерываний должно начаться только при следующей смене кадра		+		PGAsI
BNAT ⁶⁾	G	Натуральный переход к первому сплайн-кадру	м	+		PGAsI
BOOL	K	Тип данных: Значение истинности TRUE/FALSE или 1/0		+		PGAsI
BOUND	F	Проверяет, находится ли значение в пределах определенного диапазона значений. Тожество возвращает проверяемое значение.		+	+	PGAsI
BRISK ⁶⁾	G	Скачкообразное ускорение по траектории	м	+		PGAsI
BRISKA	P	Включение скачкообразного ускорения по траектории для запрограммированных осей		+	-	PGAsI
BSPLINE	G	В-сплайн	м	+		PGAsI
BTAN	G	Тангенциальный переход к первому сплайн-кадру	м	+		PGAsI
C	A	Имя оси	м/п	+		PGAsI
C2	A	Ориентация инструмента: Угол RPY или эйлеров угол	п	+		PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
C3	A	Ориентация инструмента: 3-я компонента вектора направления	п	+		PGAsl
C4	A	Ориентация инструмента: 3-я компонента вектора нормали к плоскости на начале кадра	п	+		PGAsl
C5	A	Ориентация инструмента: 3-я компонента вектора нормали к плоскости на конце кадра	п	+		PGAsl
C6	A	Ориентация инструмента: 3-я компонента вектора направления для оси вращения конуса	п	+		PGAsl
C7	A	Ориентация инструмента: 3-я компонента вектора для промежуточной ориентации на боковой поверхности конуса	п	+		PGAsl
CAC	K	Абсолютный подвод к позиции		+		PGAsl
CACN	K	Абсолютный подвод к зафиксированному в таблице значению в отрицательном направлении		+		PGAsl
CACP	K	Абсолютный подвод к зафиксированному в таблице значению в положительном направлении		+		PGAsl
CALCDAT	F	Вычисляет радиус и центр окружности из 3 или 4 точек		+	-	PGAsl
CALCPOSI	F	Проверка нарушения защищенной области, ограничения рабочего поля и программных лимитов		+	-	PGAsl
CALL	K	Косвенный вызов подпрограммы		+		PGAsl
CALLPATH	P	Программируемый маршрут поиска для вызовов подпрограмм		+	-	PGAsl
Cancel	P	Отменить модальное синхронное действие		+	-	FBSYsl
CASE	K	Условное ветвление программы		+		PGAsl
CDC	K	Прямой подвод к позиции		+		PGAsl
CDOF ⁶⁾	G	Выключить контроль столкновений	м	+		PGsl
CDOF2	G	Выключить контроль столкновений, при периферийном 3D-фрезеровании	м	+		PGsl
CDON	G	Включить контроль столкновений	м	+		PGsl
CFC ⁶⁾	G	Постоянная подача на контуре	м	+		PGsl
CFIN	G	Постоянная подача только для внутреннего изгиба, не для наружного изгиба	м	+		PGsl
CFINE	F	Присвоение точного смещения ФРЕЙМ-переменной		+	-	PGAsl
CFTCP	G	Постоянная подача в исходной точке резцов инструмента, траектория центра	м	+		PGsl
Chan	K	Спецификация сферы действия данных		+		PGAsl
CHANDATA	P	Установить номера канала для обращения к данным канала		+	-	PGAsl
CHAR	K	Тип данных: Символ ASCII		+		PGAsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
CHF	A	Фаска; значение = длина фаски	п	+		PGsI
CHKDM	F	Проверка однозначности в магазине		+	-	FBWsl
CHKDNO	F	Проверка однозначности номеров D		+	-	PGAsI
CHR	A	Фаска; значение = длина фаски в направлении движения		+		PGsI
CIC	K	Инкрементальный подвод к позиции		+		PGAsI
CIP	G	круговая интерполяция через промежуточную точку	м	+		PGsI
CLEARM	P	Сброс одной/нескольких меток для координации канала		+	+	PGAsI
CLRINT	P	Выключение прерываний		+	-	PGAsI
CMIRROR	F	отражение на оси координат		+	-	PGAsI
COARSEA	K	Конец движения при достижении "Точного останова грубого"	м	+		PGAsI
COLLPAIR	F	Проверка на отношение к соударяемой паре		+		PGAsI
COMPCAD	G	Включить функцию компрессора COMPCAD	м	+		PGAsI
COMPCURV	G	Включить функцию компрессора COMPCURV	м	+		PGAsI
COMPLETE		Управляющий оператор для выгрузки и загрузки данных		+		PGAsI
COMPOF ⁶⁾	G	Выключить сжатие кадров УП	м	+		PGAsI
COMPON	G	Включить функцию компрессора COMPON	м	+		PGAsI
COMPSURF	G	Включить функцию компрессора COMPSURF	м	+		PGAsI
CONTDCON	P	Включить декодирование контура в табличной форме		+	-	PGAsI
CONTPRON	P	Включение эталонной подготовки		+	-	PGAsI
CORROF	P	Все активные наложенные движения отключаются.		+	-	PGsI
CORRTRAFO	F	Изменение векторов смещения или векторов осей ориентации в кинематической модели станка		+	-	PGAsI
COS	F	Косинус (тригон. функция)		+	+	PGAsI
COUPDEF	P	Определение структуры электронного редуктора / соединения синхронных шпинделей		+	-	PGAsI
COUPDEL	P	Удалить структуру электронного редуктора		+	-	PGAsI
COUPOF	P	Выключить структуру электронного редуктора / синхронную шпиндельную пару		+	-	PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
COUPOFS	P	Отключение структуры электронного редуктора / синхронной шпиндельной пары с остановкой ведомого шпинделя		+	-	PGAsI
COUPON	P	Включить структуру электронного редуктора / синхронную шпиндельную пару		+	-	PGAsI
COUPONC	P	Включение структуры электронного редуктора/пары синхронных шпинделей с передачей предшествующего программирования		+	-	PGAsI
COUPRES	P	Сбросить структуру электронного редуктора		+	-	PGAsI
CP ⁶⁾	G	Движение по траектории	м	+		PGAsI
CPBC	K	Базовое соединение: Критерий смены кадров		+	+	FB3sI (M3)
CPDEF	K	Базовое соединение: Создание модуля сопряжения		+	+	FB3sI (M3)
CPDEL	K	Базовое соединение: Удаление модуля сопряжения		+	+	FB3sI (M3)
CPFMOF	K	Базовое соединение: Поведение ведомой оси при полном выключении		+	+	FB3sI (M3)
CPFMON	K	Базовое соединение: Поведение ведомой оси при включении		+	+	FB3sI (M3)
CPFMSON	K	Базовое соединение: Режим синхронизации		+	+	FB3sI (M3)
CPFPOS	K	Базовое соединение: синхронная позиция ведомой оси		+	+	FB3sI (M3)
CPFRS	K	Базовое соединение: Исходная система координат		+	+	FB3sI (M3)
CPLA	K	Базовое соединение: Определение ведущей оси		+	-	FB3sI (M3)
CPLCTID	K	Базовое соединение: Номер таблицы кривых		+	+	FB3sI (M3)
CPLDEF	K	Базовое соединение: Определение ведущей оси и создание модуля сопряжения		+	+	FB3sI (M3)
CPLDEL	K	Базовое соединение: Удаление ведущей оси модуля сопряжения		+	+	FB3sI (M3)
CPLDEN	K	Базовое соединение: Знаменатель коэффициента связи		+	+	FB3sI (M3)
CPLINSC	K	Базовое соединение: Коэффициент масштабирования для входного значения ведущей оси		+	+	FB3sI (M3)
CPLINTR	K	Базовое соединение: Значение смещения для входного значения ведущей оси		+	+	FB3sI (M3)
CPLNUM	K	Базовое соединение: Числитель коэффициента связи		+	+	FB3sI (M3)
CPLOF	K	Базовое соединение: Выключение ведущей оси модуля сопряжения		+	+	FB3sI (M3)

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
CPLON	K	Базовое соединение: Включение ведущей оси модуля сопряжения		+	+	FB3sl (M3)
CPLOUTSC	K	Базовое соединение: Коэффициент масштабирования для выходного значения соединения		+	+	FB3sl (M3)
CPLOUTTR	K	Базовое соединение: Значение смещения для выходного значения соединения		+	+	FB3sl (M3)
CPLPOS	K	Базовое соединение: синхронная позиция ведущей оси		+	+	FB3sl (M3)
CPLSETVAL	K	Базовое соединение: Источник соединения		+	+	FB3sl (M3)
CPMALARM	K	Базовое соединение: Блокировка вывода спец. относящихся к соединению аварийных сообщений		+	+	FB3sl (M3)
CPMBRAKE	K	Базовое соединение: Поведение ведомой оси при определенных сигналах и командах останова		+	-	FB3sl (M3)
CPMPRT	K	Базовое соединение: Поведение соединения при запуске программы обработки детали под поиск кадра через тестирование программы		+	+	FB3sl (M3)
CPMRESET	K	Базовое соединение: Поведение соединения при RESET		+	+	FB3sl (M3)
CPMSTART	K	Базовое соединение: Поведение соединения при запуске программы обработки детали		+	+	FB3sl (M3)
CPMVDI	K	Базовое соединение: Реакция ведомой оси на определенные интерфейсные сигналы ЧПУ/PLC		+	+	FB3sl (M3)
CPOF	K	Базовое соединение: Выключение модуля сопряжения		+	+	FB3sl (M3)
CPON	K	Базовое соединение: Включение модуля сопряжения		+	+	FB3sl (M3)
CPRECOF ⁶⁾	G	Выключение программируемой точности контура	M	+		PGAsl
CPRECON	G	Включение программируемой точности контура	M	+		PGAsl
CPRES	K	Базовое соединение: Активирует сконфигурированные данные соединения синхронных шпинделей		+	-	FB3sl (M3)
CPROT	P	Включить/выключить спец. для канала защищенную область		+	-	PGAsl
CPROTDEF	P	Определение специфической для канала защищенной области		+	-	PGAsl
CPSETTYPE	K	Базовое соединение: Тип соединения		+	+	FB3sl (M3)
CPSYNCOF	K	Базовое соединение: Пороговое значение синхронизации позиций "Грубо"		+	+	FB3sl (M3)
CPSYNCOF2	K	Базовое соединение: Пороговое значение синхронизации позиций "Грубо" 2		+	+	FB3sl (M3)

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
CPSYNCOV	K	Базовое соединение: Пороговое значение синхронизации скорости "Грубо"		+	+	FB3sl (M3)
CPSYNFIP	K	Базовое соединение: Пороговое значение синхронизации позиций "Точно"		+	+	FB3sl (M3)
CPSYNFIP2	K	Базовое соединение: Пороговое значение синхронизации позиций "Точно" 2		+	+	FB3sl (M3)
CPSYNFIV	K	Базовое соединение: Пороговое значение синхронизации скорости "Точно"		+	+	FB3sl (M3)
CR	A	Радиус окружности	п	+		PGsl
CROT	F	Вращение актуальной системы координат		+	-	PGAsl
CROTS	F	Программируемые вращения фрейма с пространственными углами (вращение в указанных осях)	п	+	-	PGsl
CRPL	F	Вращение фрейма в произвольной плоскости		+	-	FB1sl (K2)
CSCALE	F	Коэффициент масштабирования для нескольких осей		+	-	PGAsl
CSPLINE	F	Кубический сплайн	м	+		PGAsl
CT	G	Окружность с тангенциальным переходом	м	+		PGsl
CTAB	F	Определить позицию ведомой оси на основе позиции ведущей оси из таблицы кривых		+	+	PGAsl
CTABDEF	P	Включить определение таблицы		+	-	PGAsl
CTABDEL	P	Удалить таблицу кривых		+	-	PGAsl
CTABEND	P	Выключить определение таблицы		+	-	PGAsl
CTABEXISTS	F	Проверяет таблицу кривых с номером n		+	+	PGAsl
CTABFNO	F	Число еще возможных таблиц кривых в памяти		+	+	PGAsl
CTABFPOL	F	Число еще возможных полиномов в памяти		+	+	PGAsl
CTABFSEG	F	Число еще возможных сегментов кривых в памяти		+	+	PGAsl
CTABID	F	Выводит номер n-ной таблицы кривых		+	+	PGAsl
CTABINV	F	Определить позицию ведущей оси на основе позиции ведомой оси из таблицы кривых		+	+	PGAsl
CTABISLOCK	F	Возвращает состояние блокировки таблицы кривых с номером n		+	+	PGAsl
CTABLOCK	P	Удаление и перезапись, заблокировать		+	+	PGAsl
CTABMEMTYP	F	Возвращает память, в которой создана таблица кривых с номером n.		+	+	PGAsl
CTABMPOL	F	Число макс. возможных полиномов в памяти		+	+	PGAsl
CTABMSEG	F	Число макс. возможных сегментов кривых в памяти		+	+	PGAsl
CTABNO	F	Число определенных таблиц кривых в SRAM или DRAM		+	+	FB3sl (M3)
CTABNOMEM	F	Число определенных таблиц кривых в SRAM или DRAM		+	+	PGAsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
STABPERIOD	F	Возвращает периодичность таблицы кривых с номером n		+	+	PGAsI
STABPOL	F	Число уже использованных полиномов в памяти		+	+	PGAsI
STABPOLID	F	Число использованных таблицей кривых с номером n криволинейных полиномов		+	+	PGAsI
STABSEG	F	Число уже использованных сегментов кривой в памяти		+	+	PGAsI
STABSEGID	F	Число использованных таблицей кривых с номером n сегментов кривой		+	+	PGAsI
STABSEV	F	Выводит конечное значение ведомой оси сегмента таблицы кривых		+	+	PGAsI
STABSSV	F	Выводит стандартное значение ведомой оси сегмента таблицы кривых		+	+	PGAsI
STABTEP	F	Выводит значение ведущей оси на конце таблицы кривых		+	+	PGAsI
STABTEV	F	Выводит значение ведомой оси на конце таблицы кривых		+	+	PGAsI
STABTMAX	F	Выводит макс. значение ведомой оси таблицы кривых		+	+	PGAsI
STABTMIN	F	Выводит мин. значение ведомой оси таблицы кривых		+	+	PGAsI
STABTSP	F	Выводит значение ведущей оси на начале таблицы кривых		+	+	PGAsI
STABTSV	F	Выводит значение ведомой оси на начале таблицы кривых		+	+	PGAsI
STABUNLOCK	P	Снятие блокировки удаления и перезаписи		+	+	PGAsI
CTOL	A	Допуск для функций компрессора, сглаживания ориентации и типов перешлифовки	м	+		PGAsI
CTRANS	F	Смещение нулевой точки для нескольких осей		+	-	PGAsI
CUT2D ⁶⁾	G	Коррекция радиуса 2D	м	+		PGsI
CUT2DD	G	Коррекция радиуса 2½D относительно дифференциального инструмента	м	+		PGsI
CUT2DF	G	Коррекция радиуса 2D, относительно текущего фрейма (наклонная плоскость)	м	+		PGsI
CUT2DFD	G	Коррекция радиуса 2½D относительно дифференциального инструмента, относительно текущего фрейма (наклонная плоскость)	м	+		PGsI
CUT3DC	G	Коррекция радиуса 3D для периферийного фрезерования	м	+		PGAsI
CUT3DCC	G	Коррекция радиуса 3D для периферийного фрезерования с учетом ограничивающей поверхности с коррекцией радиуса 3D: Контур на поверхности обработки	м	+		PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
CUT3DCCD	G	Коррекция радиуса 3D для периферийного фрезерования с учетом ограничивающей поверхности с дифференциальным инструментом на траектории центра инструмента: Подача к ограничивающей поверхности	м	+		PGAsl
CUT3DCD	G	Коррекция радиуса 3D относительно дифференциального инструмента для периферийного фрезерования	м	+		PGAsl
CUT3DF	G	Коррекция радиуса 3D для торцового фрезерования с изменением ориентации	м	+		PGAsl
CUT3DFD	G	Коррекция радиуса 3D относительно дифференциального инструмента для торцового фрезерования с изменением ориентации	м	+		PGAsl
CUT3DFF	G	Коррекция радиуса 3D для торцового фрезерования с постоянной ориентацией. Ориентация инструмента — это определенное через G17 – G19 и в некоторых случаях повернутое через фрейм направление.	м	+		PGAsl
CUT3DFS	G	Коррекция радиуса 3D для торцового фрезерования с постоянной ориентацией. Ориентация инструмента определена через G17 – G19 и не подвергается воздействию фреймов.	м	+		PGAsl
CUTCONOF ⁶⁾	G	Выключить постоянную коррекцию радиуса	м	+		PGsl
CUTCONON	G	Включить постоянную коррекцию радиуса	м	+		PGsl
CUTMOD	A	Включение изменения данных коррекции для вращающихся инструментов (в сочетании с ориентируемыми инструментальными суппортами)	м	+		PGAsl
CUTMODK	A	Включение изменения данных коррекции для вращающихся инструментов (в сочетании с трансформациями ориентации, определенными посредством кинематических цепочек)	м	+		PGAsl
CYCLE60	C (T)	Цикл гравирования		+		PGAsl
CYCLE61	C (T)	Плоское фрезерование		+		PGAsl
CYCLE62	C (T)	Вызов контура		+		PGAsl
CYCLE63	C (T)	Фрезерование контурного кармана		+		PGAsl
CYCLE64	C (T)	Предварительное сверление контурного кармана		+		PGAsl
CYCLE70	C (T)	Резьбофрезерование		+		PGAsl
CYCLE72	C (T)	Фрезерование траектории		+		PGAsl
CYCLE76	C (T)	Фрезерование прямоугольной цапфы		+		PGAsl
CYCLE77	C (T)	Фрезерование круговой цапфы		+		PGAsl
CYCLE78	C (T)	Сверлильное резьбофрезерование		+		PGAsl
CYCLE79	C (T)	Многогранник		+		PGAsl

21.1 Операторы

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
CYCLE81	C (T)	Сверление, центрование		+		PGAsI
CYCLE82	C (T)	Сверление, зенкование		+		PGAsI
CYCLE83	C (T)	Глубокое сверление		+		PGAsI
CYCLE84	C (T)	Нарезание внутренней резьбы без компенсирующего патрона		+		PGAsI
CYCLE85	C (T)	Развертывание		+		PGAsI
CYCLE86	C (T)	Растачивание		+		PGAsI
CYCLE92	C (T)	Отрез		+		PGAsI
CYCLE95	C (T)	Съём припуска вдоль контура		+		PGAsI
CYCLE98	C (T)	Цепочка резьб		+		PGAsI
CYCLE99	C (T)	Нарезание резьбы резцом		+		PGAsI
CYCLE150	C (M)	Индикация / регистрация результатов измерения		+		BNMsl
CYCLE435	C (T)	Вычислить позицию инструмента для правки		+		PGAsI
CYCLE495	C (T)	Профилирование		+		PGAsI
CYCLE750	C (A)	Внутренний рабочий цикл для CYCLE751 ... CYCLE759 (содержит команду MMC для прямого вызова функции)		-		FB3sl (T4)
CYCLE751	C (A)	Открыть / выполнить / закрыть сеанс оптимизации		M		FB3sl (T4)
CYCLE752	C (A)	Добавить ось к сеансу оптимизации		M		FB3sl(T4)
CYCLE753	C (A)	Выбрать режим оптимизации		M		FB3sl (T4)
CYCLE754	C (A)	Добавить / удалить блок данных		M		FB3sl (T4)
CYCLE755	C (A)	Архивировать / восстановить блок данных		M		FB3sl (T4)
CYCLE756	C (A)	Активировать результаты оптимизации		M		FB3sl (T4)
CYCLE757	C (A)	Сохранить данные оптимизации		M		FB3sl (T4)
CYCLE758	C (A)	Изменить значение параметра		M		FB3sl (T4)
CYCLE759	C (A)	Считать значение параметра		M		FB3sl (T4)
CYCLE800	C (T)	Поворот		+		PGAsI
CYCLE801	C (T)	Решетка или матрица		+		PGAsI
CYCLE802	C (T)	Любые позиции		+		PGAsI
CYCLE830	C (T)	Глубокое сверление 2		+		PGAsI
CYCLE832	C (T)	High Speed Settings		+		PGAsI
CYCLE840	C (T)	Нарезание внутренней резьбы с компенсирующим патроном		+		PGAsI
CYCLE899	C (T)	Фрезерование открытой канавки		+		PGAsI
CYCLE930	C (T)	Выточка		+		PGAsI
CYCLE940	C (T)	Формы канавки		+		PGAsI
CYCLE951	C (T)	Обработка резаньем		+		PGAsI
CYCLE952	C (T)	Прорезание контура		+		PGAsI
CYCLE961	C (M)	Определение положения угла детали (внутри или снаружи) и использование как смещения нулевой точки		+		BNMsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
CYCLE971	C (M)	Калибровка измерительного щупа инструмента, измерение длины инструмента и/или радиуса инструмента (только для технологии фрезерования)		+		BNMsl
CYCLE973	C (M)	Калибровка измерительного щупа детали на поверхности детали или в пазу (только для технологии токарной обработки)		+		BNMsl
CYCLE974	C (M)	Определение нулевой точки детали в выбранной измерительной оси, коррекция инструмента с помощью 1-точечного измерения (только для технологии токарной обработки)		+		BNMsl
CYCLE976	C (M)	Калибровка измерительного щупа детали в калибровочном кольце или на калибровочной сфере полностью в рабочей плоскости или на кромке для определенной оси и направления		+		BNMsl
CYCLE977	C (M)	Определение центра плоскости, а также ширины или диаметра		+		BNMsl
CYCLE978	C (M)	Измерение положения кромки в системе координат детали		+		BNMsl
CYCLE979	C (M)	Измерение центра плоскости, измерение радиуса круговых сегментов		+		BNMsl
CYCLE982	C (M)	Калибровка измерительного щупа инструмента, измерение токарного, сверлильного и фрезероувального инструмента (только для технологии токарной обработки)		+		BNMsl
CYCLE994	C (M)	Определение нулевой точки детали в выбранной измерительной оси с помощью 2-точечного измерения (только для технологии токарной обработки)		+		BNMsl
CYCLE995	C (M)	Измерение наклона шпинделя на станке		+		BNMsl
CYCLE996	C (M)	Определение данных, релевантных для кинематических трансформаций, с использованием круговых осей		+		BNMsl
CYCLE997	C (M)	Определение центра и диаметра сферы, измерение центров трех распределенных сфер		+		BNMsl
CYCLE998	C (M)	Определение углового положения поверхности (плоскости) относительно рабочей плоскости, определение угла кромок в системе координат детали		+		BNMsl
CYCLE4071	C (T)	Продольное шлифование с подачей в точке возврата		+		PGAsl
CYCLE4072	C (T)	Продольное шлифование с подачей в точке возврата и сигналом отмены		+		PGAsl
CYCLE4073	C (T)	Продольное шлифование с непрерывной подачей		+		PGAsl

21.1 Операторы

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
CYCLE4074	C (T)	Продольное шлифование с непрерывной подачей и сигналом отмены		+		PGAsI
CYCLE4075	C (T)	Плоское шлифование с подачей в точке возврата		+		PGAsI
CYCLE4077	C (T)	Плоское шлифование с подачей в точке возврата и сигналом отмены		+		PGAsI
CYCLE4078	C (T)	Плоское шлифование с непрерывной подачей		+		PGAsI
CYCLE4079	C (T)	Плоское шлифование с прерывистой подачей		+		PGAsI

Операторы D ... F

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
D	A	Номер коррекции инструмента		+		PGsI
D0	A	При D0 коррекции для инструмента не действуют.		+		PGsI
DAC	K	Абсолютное покадровое спец. для оси программирование диаметра	c	+		PGsI
DC	K	Абсолютное указание размеров для круговых осей, прямой подвод к позиции	c	+		PGsI
DCI	K	Присвоить класс данных I (= Individual) (только SINUMERIK 828D!)		+		PGAsI
DCM	K	Присвоить класс данных M (= Manufacturer) (только SINUMERIK 828D!)		+		PGAsI
DCU	K	Присвоить класс данных U (= User) (только SINUMERIK 828D!)		+		PGAsI
DEF	K	Определение переменных		+		PGAsI
DEFAULT	K	Ветвь в ветвлении CASE		+		PGAsI
DEFINE	K	Кодовое слово для макроопределений		+		PGAsI
DELAYFSTOF	P	Определить конец области задержки останова	m	+	-	PGAsI
DELAYFSTON	P	Определить начало области задержки останова	m	+	-	PGAsI
DELDL	F	Удалить аддитивные коррекции		+	-	PGAsI
DELDTG	P	Стирание остатка пути		-	+	FBSYsI
DELETE	P	Удалить указанный файл. Имя файла может быть указано с маршрутом и идентификатором файла.		+	-	PGAsI
DELMLOWNER	F	Удалить место в собственном магазине инструмента		+	-	FBWsI
DELMRES	F	Удалить резервирование места в магазине		+	-	FBWsI
DELMT	P	Удалить мультиинструмент		+	-	FBWsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1)2)3)4)5) Объяснения см. Экспликация (Страница 914).						
DELOBJ	F	Удаление элементов кинематических цепочек, защищенных областей, элементов защищенных областей, соударяемых пар и данных трансформаций		+		PGAsl
DELT	P	Удалить инструмент		+	-	FBWsl
DELTC	P	Удалить блок данных инструментального суппорта		+	-	FBWsl
DELTOOLENV	F	Удалить блоки данных для описания инструментального окружения		+	-	PGAsl
DIACYCOFA	K	Спец. для оси модальное программирование диаметра: ВЫКЛ в циклах	м	+		FB1sl (P1)
DIAM90	G	Программирование диаметра для G90, программирование радиуса для G91	м	+		PGAsl
DIAM90A	K	Спец. для оси модальное программирование диаметра для G90 и AC, программирование радиуса для G91 и IC	м	+		PGsl
DIAMCHAN	K	Передача всех осей из MD функций осей в состояние канала программирования диаметра		+		PGsl
DIAMCHANA	K	Применение состояния канала программирования диаметра		+		PGsl
DIAMCYCOF	G	Спец. для канала программирование диаметра: ВЫКЛ в циклах	м	+		FB1sl (P1)
DIAMOF ⁶⁾	G	Программирование диаметра: ВЫКЛ Первичная установка см. Изготовитель станка	м	+		PGsl
DIAMOFA	K	Спец. для оси модальное программирование диаметра: ОТКЛ Первичная установка см. Изготовителя станка	м	+		PGsl
DIAMON	G	Программирование диаметра: ВКЛ	м	+		PGsl
DIAMONA	K	Спец. для оси модальное программирование диаметра: ВКЛ Активация см. Изготовитель станка	м	+		PGsl
DIC	K	Относительное покадровое спец. для оси программирование диаметра	с	+		PGsl
DILF	A	Путь обратного хода (длина)	м	+		PGsl
Disable	P	Прерывание ВЫКЛ.		+	-	PGAsl
DISC	A	Выступ переходного круга коррекции радиуса инструмента	м	+		PGsl
DISCL	A	Расстояние от конечной точки быстрого движения подачи до плоскости обработки		+		PGsl
DISPLOF	PA	Подавление актуальной индикации кадра		+		PGAsl
DISPLON	PA	Отменить подавление актуальной индикации кадра		+		PGAsl
DISPR	A	Рассогласование траектории Repos	с	+		PGAsl
DISR	A	Интервал Repos	с	+		PGAsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Объяснения см. Экспликация (Страница 914).						
DISRP	A	Расстояние от плоскости отвода до плоскости обработки при мягком подводе и отводе		+		PGsl
DITE	A	Выход резьбы	м	+		PGsl
DITS	A	Вход резьбы	м	+		PGsl
DIV	K	Целочисленное деление		+		PGAsl
DL	A	Выбрать зависящую от места аддитивную коррекцию инструмента (DL, суммарная отладочная коррекция)	м	+		PGAsl
DO	A	Кодовое слово для синхронного действия, запускает действие при выполненном условии		-	+	FBSYsl
DRFOF	P	Выключение смещений маховичком (DRF)	м	+	-	PGsl
DRIVE	G	Ускорение по траектории в зависимости от скорости	м	+		PGAsl
DRIVEA	P	Включение ломаной характеристики ускорения для запрограммированных осей		+	-	PGAsl
DYNFINISH	G	Динамика для чистовой обработки точной	м	+		PGAsl
DYNNORM ⁶⁾	G	Обычная динамика	м	+		PGAsl
DYNPOS	G	Динамика для режима позиционирования, нарезание внутренней резьбы	м	+		PGAsl
DYNROUGH	G	Динамика для черновой обработки	м	+		PGAsl
DYNSEMIFIN	G	Динамика для чистовой обработки	м	+		PGAsl
DZERO	P	Обозначает все номера D блока TO как недействительные		+	-	PGAsl
EAUTO	G	Определение последнего сегмента сплайна через последние 3 точки	м	+		PGAsl
EGDEF	P	Определение электронного редуктора		+	-	PGAsl
EGDEL	P	Удалить определение соединения для ведомой оси		+	-	PGAsl
EGOFC	P	Непрерывное выключение электронного редуктора		+	-	PGAsl
EGOFS	P	Селективное выключение электронного редуктора		+	-	PGAsl
EGON	P	Включение электронного редуктора		+	-	PGAsl
EGONSYN	P	Включение электронного редуктора		+	-	PGAsl
EGONSYNE	P	Включение электронного редуктора, с задаче режима пуска		+	-	PGAsl
ELSE	K	Ветвление программы, если условие IF не выполнено		+		PGAsl
Enable	P	Прерывание ВКЛ.		+	-	PGAsl
ENAT ⁶⁾	G	Натуральный криволинейный переход к следующему кадру перемещения	м	+		PGAsl
ENDFOR	K	Конечная строка цикла вычисления FOR		+		PGAsl
ENDIF	K	Конечная строка ветвления IF		+		PGAsl
ENDLABEL	K	Метка конца для повторений программы обработки детали через REPEAT		+		PGAsl, FB1sl (K1)

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Объяснения см. Экспликация (Страница 914).						
ENDLOOP	K	Конечная строка бесконечного программного цикла LOOP		+		PGAsl
ENDPROC	K	Конечная строка программы с начальной строкой PROC		+		
ENDWHILE	K	Конечная строка цикла WHILE		+		PGAsl
ESRR	P	Спараметрировать автономный отвода привода ESR в приводе		+		PGAsl
ESRS	P	Спараметрировать автономный останов привода ESR в приводе		+		PGAsl
ETAN	G	Тангенциальный криволинейный переход к следующему кадру перемещения на начале сплайна	м	+		PGAsl
EVERY	K	Выполнить синхронное действие при переходе условия с FALSE на TRUE		-	+	FBSYsl
Ex	K	Кодовое значение для присвоения значения при экспоненциальной форме записи		+		PGAsl
EXECSTRING	P	Передача переменной String с исполняемой строкой программы обработки детали		+	-	PGAsl
EXECTAB	P	Обработать элемент из таблицы движений		+	-	PGAsl
EXECUTE	P	Выполнение программы ВКЛ		+	-	PGAsl
exp	F	Показательная функция ex		+	+	PGAsl
EXTCALL	A	Выполнить внешнюю подпрограмму		+	+	PGAsl
EXTCLOSE	P	Для записи закрыть открытое внешнее устройство/файл		+	-	PGAsl
EXTERN	K	Публикация подпрограммы с передачей параметров		+		PGAsl
EXTOPEN	P	Открыть внешнее устройство/файл для канала для записи		+	-	PGAsl
F	A	Значение подачи (вместе с G4 с F программируется и время ожидания)		+	+	PGsl
FA	K	Осевая подача	м	+	+	PGsl
FAD	A	Подача на ход для мягкого подвода и отвода		+		PGsl
false	K	Логическая постоянная: неверно		+	+	PGAsl
FB	A	Покадровая подача		+		PGsl
FCTDEF	P	Определение функции полинома		+	-	PGAsl
FCUB	G	подача изменяется по кубическому сплайну	м	+		PGAsl
FD	A	Подача по траектории для наложения маховичка	с	+		PGsl
FDA	K	Осевая подача для наложения маховичка	с	+		PGsl
FENDNORM ⁶⁾	G	Замедление на углах ВЫКЛ	м	+		PGAsl
FFWOF ⁶⁾	G	Предупреждение ВЫКЛ	м	+		PGAsl
FFWON	G	Предупреждение Вкл	м	+		PGAsl

21.1 Операторы

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Объяснения см. Экспликация (Страница 914).						
FGREF	K	Исходный радиус для круговых осей или коэффициенты соотношения траекторий для осей ориентации (векторная интерполяция)	м	+		PGsl
FGROUP	P	Определение оси(ей) с подачей по траектории		+	-	PGsl
FI	K	Параметры для доступа к данным фрейма: Точное смещение		+		PGAsl
FIFOCTRL	G	Управление буфером предварительной обработки	м	+		PGAsl
FILEDATE	P	Выводит дату последнего обращения по записи к файлу		+	-	PGAsl
FILEINFO	P	Выводит сумму FILEDATE, FILESIZE, FILESTAT и FILETIME		+	-	PGAsl
FILESIZE	P	Выводит актуальный размер файла		+	-	PGAsl
FILESTAT	P	Выводит состояние файла прав чтения, записи, исполнения, индикации, удаления (rwxsd)		+	-	PGAsl
FILETIME	P	Выводит время последнего обращения по записи к файлу		+	-	PGAsl
FINEA	K	Конец движения при достижении "Точного останова точного"	м	+		PGAsl
FL	K	Предельная скорость для синхронных осей	м	+		PGsl
FLIN	G	подача изменяется линейно	м	+		PGAsl
FMA	K	Несколько осевых подач	м	+		PGsl
FNORM ⁶⁾	G	обычная подача по DIN66025	м	+		PGAsl
FOC	K	Действующее покадровое ограничение момента/силы	с	-	+	FBSYsl
FOCOF	K	Выключение модального ограничения момента/силы	м	-	+	FBSYsl
FOCON	K	Включение модального ограничения момента/силы	м	-	+	FBSYsl
FOR	K	Цикл вычисления с фиксированным числом проходов		+		PGAsl
FP	A	Фиксированная точка: номер фикс. точки, к которой осуществляется подвод	с	+		PGsl
FPO	K	Запрограммированная через полином характеристика подачи		+		PGAsl
FPR	P	Обозначение круговой оси		+	-	PGsl
FPRAOF	P	Выключение окружной подачи		+	-	PGsl
FPRAON	P	Включение окружной подачи		+	-	PGsl
FRAME	K	Тип данных для определения систем координат		+		PGAsl
FRC	A	Подача для радиуса и фаски	с	+		PGsl
FRCM	A	Модальная подача для радиуса и фаски	м	+		PGsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
¹⁾²⁾³⁾⁴⁾⁵⁾ Объяснения см. Экспликация (Страница 914).						
From	K	Операция выполняется, если условие выполнено один раз и пока активно синхронное действие		-	+	FBSYsl
FTOC	P	Изменить точную коррекцию инструмента		-	+	FBSYsl
FTOCOF ⁶⁾	G	действующая online точная коррекция инструмента ВЫКЛ	м	+		PGAsl
FTOCON	G	действующая online точная коррекция инструмента ВКЛ	м	+		PGAsl
FXS	K	Наезд на жесткий упор вкл	м	+	+	PGsl
FXST	K	Граница момента для наезда на жесткий упор	м	+	+	PGsl
FXSW	K	Окно контроля для наезда на жесткий упор		+	+	PGsl
FZ	K	Подача на зуб	м	+		PGsl

Операторы G ... L

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
G0	G	Линейная интерполяция с ускоренным ходом (движение ускоренного хода)	м	+		PGsl
G1 ⁶⁾	G	Линейная интерполяция с подачей	м	+		PGsl
G2	G	Круговая интерполяция по часовой стрелке	м	+		PGsl
G3	G	Круговая интерполяция против часовой стрелки	м	+		PGsl
G4	G	Время ожидания, заранее определено по времени	с	+		PGsl
G5	G	Шлифование с врезанием с угловой подачей	с	+		PGAsl
G7	G	Компенсационное движение при шлифовании с врезанием с угловой подачей	с	+		PGAsl
G9	G	Точный останов - снижение скорости	с	+		PGsl
G17 ⁶⁾	G	Выбор рабочей плоскости X/Y	м	+		PGsl
G18	G	Выбор рабочей плоскости Z/X	м	+		PGsl
G19	G	Выбор рабочей плоскости Y/Z	м	+		PGsl
G25	G	Нижнее ограничение рабочего поля	с	+		PGsl
G26	G	Верхнее ограничение рабочего поля	с	+		PGsl
G33	G	Резьбонарезание с постоянным шагом	м	+		PGsl
G34	G	Резьбонарезание с линейной увеличивающимся шагом	м	+		PGsl
G35	G	Резьбонарезание с линейной уменьшающимся шагом	м	+		PGsl
G40 ⁶⁾	G	Коррекция радиуса инструмента ВЫКЛ.	м	+		PGsl
G41	G	Коррекция радиуса инструмента слева от контура	м	+		PGsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
G42	G	Коррекция радиуса инструмента справа от контура	м	+		PGsl
G53	G	Подавление актуального смещения нулевой точки (покадрово)	с	+		PGsl
G54	G	1-е устанавливаемое смещение нулевой точки	м	+		PGsl
G55	G	2-е настраиваемое смещение нулевой точки	м	+		PGsl
G56	G	3-е настраиваемое смещение нулевой точки	м	+		PGsl
G57	G	4-е настраиваемое смещение нулевой точки	м	+		PGsl
G58 (840D sl)	G	Абсолютное программируемое смещение нулевой точки (грубое смещение)	с	+		PGsl
G58 (828D)	G	5-е настраиваемое смещение нулевой точки	м	+		PGsl
G59 (840D sl)	G	Аддитивное программируемое смещение нулевой точки (точное смещение)	с	+		PGsl
G59 (828D)	G	6-е настраиваемое смещение нулевой точки	м	+		PGsl
G60 ⁶⁾	G	Точный останов - снижение скорости	м	+		PGsl
G62	G	Задержка на внутренних углах при активной коррекции радиуса инструмента (G41, G42)	м	+		PGasl
G63	G	Нарезание внутренней резьбы с компенсирующим патроном	с	+		PGsl
G64	G	Режим управления траекторией	м	+		PGsl
G70	G	Дюймовое указание размеров для геометрических данных (длины)	м	+	+	PGsl
G71 ⁶⁾	G	Метрическое указание размеров для геометрических данных (длины)	м	+	+	PGsl
G74	G	Движение к точке реферирования	с	+		PGsl
G75	G	Подвод к базисной точке	с	+		PGsl
G90 ⁶⁾	G	Абсолютное указание размера	м/с	+		PGsl
G91	G	Указание составного размера	м/с	+		PGsl
G93	G	Обратная по времени подача 1/мин	м	+		PGsl
G94 ⁶⁾	G	Линейная подача F в мм/мин или дюймов/мин и градусах/мин	м	+		PGsl
G95	G	Окружная подача F в мм/об или дюймов/оборот	м	+		PGsl
G96	G	Окружная подача (как у G95) и постоянная скорость резания	м	+		PGsl
G97	G	Окружная подача и постоянная частота вращения шпинделя (постоянная скорость резания ВЫКЛ)	м	+		PGsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
G110	G	Программирование полюса относительно последней запрограммированной заданной позиции	c	+		PGsl
G111	G	Программирование полюса относительно нулевой точки актуальной системы координат детали	c	+		PGsl
G112	G	Программирование полюса относительно последнего действующего полюса	c	+		PGsl
G140 ⁶⁾	G	Направление подвода SAR установлено через G41/G42	m	+		PGsl
G141	G	Направление подвода SAR слева от контура	m	+		PGsl
G142	G	Направление подвода SAR справа от контура	m	+		PGsl
G143	G	Направление подвода SAR зависимое от касательных	m	+		PGsl
G147	G	Мягкий подвод по прямой	c	+		PGsl
G148	G	Мягкий отвод по прямой	c	+		PGsl
G153	G	Подавление актуальных фреймов включая базовый фрейм	c	+		PGsl
G247	G	Мягкий подвод по четверти круга	c	+		PGsl
G248	G	Мягкий отвод по четверти круга	c	+		PGsl
G290 ⁶⁾	G	Переключение на режим SINUMERIK ВКЛ.	m	+		FBWsl
G291	G	Переключение на режим ISO2/3 ВКЛ.	m	+		FBWsl
G331	G	Нарезание внутренней резьбы без компенсирующего патрона, положительный шаг, правый ход	m	+		PGsl
G332	G	Нарезание внутренней резьбы без компенсирующего патрона, отрицательный шаг, левый ход	m	+		PGsl
G335	G	Нарезание резьбы с выпуклым профилем по часовой стрелке	m	+		PGsl
G336	G	Нарезание резьбы с выпуклым профилем против часовой стрелки	m	+		PGsl
G340 ⁶⁾	G	Кадр подвода пространственный (по глубине и в плоскости одновременно (спираль))	m	+		PGsl
G341	G	Подача сначала в вертикальной оси (z), потом подвод в плоскости	m	+		PGsl
G347	G	Мягкий подвод по полукругу	c	+		PGsl
G348	G	Мягкий отвод по полукругу	c	+		PGsl
G450 ⁶⁾	G	Переходная окружность	m	+		PGsl
G451	G	Точка пересечения эквидистант	m	+		PGsl
G460 ⁶⁾	G	Включение контроля столкновения для кадра подвода и отвода	m	+		PGsl
G461	G	Вставка окружности в кадре КРИ	m	+		PGsl
G462	G	Вставка прямой в кадре КРИ	m	+		PGsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
G500 ⁶⁾	G	Отключение всех устанавливаемых фреймов, базовые фреймы активны	м	+		PGsl
G505 ... G599	G	5. ... 99. Устанавливаемое смещение нулевой точки	м	+		PGsl
G601 ⁶⁾	G	Смена кадра при точном останове точном	м	+		PGsl
G602	G	Смена кадра при точном останове грубом	м	+		PGsl
G603	G	Смена кадра при конце кадра IPO	м	+		PGsl
G621	G	Замедление на всех углах	м	+		PGAsl
G641	G	Режим управления траекторией с перешлифовкой по критерию пути (= программируемый интервал перешлифовки)	м	+		PGsl
G642	G	Режим управления траекторией с перешлифовкой с соблюдением определенных допусков	м	+		PGsl
G643	G	Режим управления траекторией с перешлифовкой с соблюдением определенных допусков (внутри кадра)	м	+		PGsl
G644	G	Режим управления траекторией с перешлифовкой с макс. возможной динамикой	м	+		PGsl
G645	G	Режим управления траекторией с перешлифовкой углов с тангенциальными переходами кадров с соблюдением определенных допусков	м	+		PGsl
G700	G	Дюймовое указание размеров для геометрических и технологических данных (длины, подача)	м	+	+	PGsl
G710 ⁶⁾	G	Метрическое указание размеров для геометрических и технологических данных (длины, подача)	м	+	+	PGsl
G810 ⁶⁾ , ..., G819	G	Зарезервированная для пользователя OEM G-группа		+		PGAsl
G820 ⁶⁾ , ..., G829	G	Зарезервированная для пользователя OEM G-группа		+		PGAsl
G931	G	Задача подачи через время перемещения, отключение постоянной скорости движения по траектории	м	+		
G942	G	Линейная подача и постоянная скорость резания или "замораживание" частоты вращения шпинделя	м	+		
G952	G	Окружная подача и постоянная скорость резания или "замораживание" частоты вращения шпинделя	м	+		
G961	G	Линейная подача (как у G94) и постоянная скорость резания	м	+		PGsl
G962	G	Линейная подача или окружная подача и постоянная скорость резания	м	+		PGsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
G971	G	Линейная подача и постоянная частота вращения шпинделя (постоянная скорость резания ВЫКЛ)	м	+		PGsl
G972	G	Линейная подача или окружная подача и постоянная частота вращения шпинделя (постоянная скорость резания ВЫКЛ)	м	+		PGsl
G973	G	Окружная подача без ограничения частоты вращения шпинделя и постоянная частота вращения шпинделя (G97 без LIMS для режима ISO)	м	+		PGsl
GEOAX	P	Согласовать с гео-осями 1 - 3 новые оси канала		+	-	PGAsl
GET	P	Переход разрешенной оси между каналами		+	+	PGAsl
GETACTT	F	Определяет активный инструмент из группы инструментов с одинаковым именем		+	-	FBWsl
GETACTTD	F	Определяет для абсолютного номера D соответствующий номер T		+	-	PGAsl
GETD	P	Прямой переход оси между каналами		+	-	PGAsl
GETDNO	F	Выводит номер D резца (CE) инструмента (T)		+	-	PGAsl
GETEXET	P	Чтение установленного номера T		+	-	FBWsl
GETFREELOC	P	Поиск свободного места в магазинах для данного инструмента		+	-	FBWsl
GETSELT	P	Вывести предварительно выбранный номер T		+	-	FBWsl
GETT	F	Определить номер T для имени инструмента		+	-	FBWsl
GETTCOR	F	Считать длину или компоненты длины инструмента		+	-	PGAsl
GETTENV	F	Чтение номеров T, D и DL		+	-	PGAsl
GETVARAP	F	Чтение права доступа к системной переменной/переменной пользователя		+	-	PGAsl
GETVARDFT	F	Чтение стандартного значения системной переменной/переменной пользователя		+	-	PGAsl
GETVARLIM	F	Чтение предельных значений системной переменной/переменной пользователя		+	-	PGAsl
GETVARPHU	F	Чтение физической единицы системной переменной/переменной пользователя		+	-	PGAsl
GETVARTYP	F	Чтение типа данных системной переменной/переменной пользователя		+	-	PGAsl
GFRAME0 ... GFRAME100	G	Активация фрейма шлифования <n> для сохранения данных в канале	м	+		PGsl
GOTO	K	Оператор перехода сначала вперед, потом назад (сначала в направлении конца программы, а после к началу программы)		+		PGAsl
GOTOB	K	Оператор перехода назад (в направлении начала программы)		+		PGAsl

21.1 Операторы

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
GOTOC	K	Как GOTO, но подавление ошибки 14080 "Цель перехода не найдена"		+		PGAsI
GOTOF	K	Оператор перехода вперед (в направлении конца программы)		+		PGAsI
GOTOS	K	Возврат в начало программы		+		PGAsI
GP	K	Кодовое слово для косвенного программирования атрибутов позиции		+		PGAsI
GROUP_ADDEND	C (T)	Конец – дополнение подведения		+		PGAsI
GROUP_BEGIN	C (T)	Начало – блок программы		+		PGAsI
GROUP_END	C (T)	Конец – блок программы		+		PGAsI
GWPSOF	P	Отмена постоянной окружной скорости круга (SUG)	c	+	-	PGsI
GWPSON	P	Выбор постоянной окружной скорости круга (SUG)	c	+	-	PGsI
H...	A	Вывод вспомогательных функций на PLC		+	+	PGsI/FB1sI (H2)
HOLES1	C (T)	Ряд отверстий		+		PGAsI
HOLES2	C (T)	Окружность отверстий		+		PGAsI
I	A	Параметры интерполяции	c	+		PGsI
I1	A	Координата промежуточной точки	c	+		PGsI
IC	K	Ввод составного размера	c	+		PGsI
ICYCOF	P	Выполнить все кадры технологического цикла после ICYCOF в одном такте IPO		+	+	FBSYsI
ICYCON	P	Выполнить каждый кадр технологического цикла после ICYCON в отдельном такте IPO		+	+	FBSYsI
ID	K	Обозначение для модальных синхронных действий	m	-	+	FBSYsI
IDS	K	Обозначение для модальных статических синхронных действий		-	+	FBSYsI
IF	K	Ввод условного перехода в программе обработки детали / технологическом цикле		+	+	PGAsI
INDEX	F	Определить индекс символа во входной строке		+	-	PGAsI
INICF	K	Инициализация переменных при NEWCONF		+		PGAsI
INIPO	K	Инициализация переменных при PowerOn		+		PGAsI
INIRE	K	Инициализация переменных при Reset		+		PGAsI
INIT	P	Выбор определенной программы ЧПУ для выполнения в определенном канале		+	-	PGAsI
INITIAL		Создание файла INI по всем областям		+		PGAsI
INT	K	Тип данных: Целочисленное значение со знаком		+		PGAsI
INTERSEC	F	Вычислить точку пересечения между двумя элементами контура		+	-	PGAsI
INVCCW	G	Движение по эвольвенте, против часовой стрелки	m	+		PGsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
INVCW	G	Движение по эвольвенте, по часовой стрелке	м	+		PGsl
INVFRAME	F	Вычислить из фрейма инверсный фрейм		+	-	FB1sl (K2)
IP	K	Переменный параметр интерполяции		+		PGAsl
IPOBRKA	P	Критерий движения от точки применения рампы торможения	м	+	+	
IPOENDA	K	Конец движения при достижении "IPO-Stop"	м	+		PGAsl
IPTRLOCK	P	"Заморозить" начало непригодного для поиска сегмента программы на следующем кадре функций станка.	м	+	-	PGAsl
IPTRUNLOCK	P	Установить конец непригодного для поиска сегмента программы на актуальный на момент прерывания кадр.	м	+	-	PGAsl
IR	A	Координата центра окружности (направление X) при нарезании резьбы с выпуклым профилем		+		PGsl
ISAXIS	F	Проверить, указанная как параметр геосось это 1		+	-	PGAsl
ISD	A	Глубина врезания	м	+		PGAsl
ISFILE	F	Проверить, есть ли файл в прикладной памяти ЧПУ		+	-	PGAsl
ISNUMBER	F	Проверить, может ли входная строка быть преобразована в число		+	-	PGAsl
ISOCALL	K	Косвенный вызов запрограммированной на языке ISO программы		+		PGAsl
ISVAR	F	Проверить, содержит ли передаваемый параметр известную в ЧПУ переменную		+	-	PGAsl
J	A	Параметры интерполяции	с	+		PGsl
J1	A	Координата промежуточной точки	с	+		PGsl
JERKA	P	Активировать установленные через MD параметры ускорения для запрограммированных осей		+	-	
JERKLIM	K	Уменьшение или увеличение макс. осевого рывка	м	+		PGAsl
JERKLIMA	K	Уменьшение или увеличение макс. осевого рывка	м	+	+	PGAsl
JR	A	Координата центра окружности (направление Y) при нарезании резьбы с выпуклым профилем		+		PGsl
K	A	Параметры интерполяции	с	+		PGsl
K1	A	Координата промежуточной точки	с	+		PGsl
KONT	G	Обход контура при коррекции инструмента	м	+		PGsl
KONTC	G	Подвод/отвод с полиномом с постоянным изгибом	м	+		PGsl
KONTT	G	Подвод/отвод с полиномом с постоянным касанием	м	+		PGsl

21.1 Операторы

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
KR	A	Координата центра окружности (направление Z) при нарезании резьбы с выпуклым профилем		+		PGsI
I	A	Номер подпрограммы	c	+	+	PGAsI
LEAD	A	Угол предварения 1. Ориентация инструмента 2. Полином ориентации	m	+		PGAsI
LEADOF	P	Осевое соединение по главному значению ВЫКЛ		+	+	PGAsI
LEADON	P	Осевое соединение по главному значению ВКЛ		+	+	PGAsI
LENTOAX	F	Выводит информацию по согласованию длин инструмента L1, L2 и L3 активного инструмента с абсциссой, ординатой и аппликатой		+	-	PGAsI
LFOF ⁶⁾	G	Быстрый обратный ход для резьбонарезания ВЫКЛ	m	+		PGsI
LFON	G	Быстрый обратный ход для резьбонарезания ВКЛ	m	+		PGsI
LFPOS	G	Обратный ход обозначенной с POLFMASK или POLFMLIN оси на запрограммированную с POLF абсолютную позицию оси	m	+		PGsI
LFTXT ⁶⁾	G	Плоскость движения обратного хода при быстром отводе определяется из касательной к траектории и актуального направления инструмента	m	+		PGsI
LFWP	G	Плоскость движения обратного хода при быстром отводе определяется через актуальную рабочую плоскость (G17/G18/G19)	m	+		PGsI
LIFTFAST	K	Быстрый отвод		+		PGsI
LIMS	K	Ограничение частоты вращения при G96/G961 и G97	m	+		PGsI
LLI	K	Нижнее предельное значение переменной		+		PGAsI
LN	F	Натуральный логарифм		+	+	PGAsI
LOCK	P	Блокировать синхронное действие с ID (остановить технологический цикл)		-	+	FBSYsI
LONGHOLE	C (T)	Продольный паз		+		PGAsI
LOOP	K	Начало бесконечного цикла		+		PGAsI

Операторы M ... R

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
M0		Запрограммированная остановка		+	+	PGsI
M1		Остановка по выбору		+	+	PGsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
M2		Конец главной программы (как M30)		+	+	PGsl
M3		Правое направление вращения шпинделя		+	+	PGsl
M4		Левое направление вращения шпинделя		+	+	PGsl
M5		Остановка шпинделя		+	+	PGsl
M6		Смена инструмента		+	+	PGsl
M17		Конец подпрограммы		+	+	PGsl
M19		Позиционирование шпинделя на введенную в SD43240 позицию		+	+	PGsl
M30		Конец главной программы (как M2)		+	+	PGsl
M40		Автоматическое переключение редуктора		+	+	PGsl
M41 ... M45		Ступень редуктора 1 ... 5		+	+	PGsl
M70		Переход в осевой режим		+	+	PGsl
MASLDEF	P	Определить структуру осей Master/Slave		+	+	PGAsl
MASLDEL	P	Разъединить структуру осей Master/Slave и удалить определение структуры		+	+	PGAsl
MASLOF	P	Отключение временного соединения		+	+	PGAsl
MASLOFS	P	Отключение временного соединения с автоматическим остановом оси Slave		+	+	PGAsl
MASLON	P	Включение временного соединения		+	+	PGAsl
MATCH	F	Поиск строки в строке		+	-	PGAsl
MAXVAL	F	Большее значение двух переменных (арифм. функция)		+	+	PGAsl
MCALL	K	Модальный вызов подпрограммы		+		PGAsl
MEAC	K	Осевое непрерывное измерение без стирания остатка пути	п	+	+	PGAsl
MEAFRAME	F	Вычисление фрейма из точек измерения		+	-	PGAsl
MEAS	A	Измерение со стиранием остатка пути	п	+		PGAsl
MEASA	K	Осевое измерение со стиранием остатка пути	п	+	+	PGAsl
MEASURE	F	Метод вычислений для измерения детали и инструмента		+	-	FB1sl (M5)
MEAW	A	Измерение без стирания остатка пути	п	+		PGAsl
MEAWA	K	Осевое измерение без стирания остатка пути	п	+	+	PGAsl
MI	K	Доступ к данным фрейма: Отражение		+		PGAsl
MINDEX	F	Определить индекс символа во входной строке		+	-	PGAsl
MINVAL	F	Меньшее значение двух переменных (арифм. функция)		+	+	PGAsl
MIRROR	G	Программируемое отражение	п	+		PGAsl
MMC	P	Интерактивный вызов диалоговых окон на HMI из программы обработки детали		+	-	PGAsl
MOD	K	Выделение дробной части		+		PGAsl

21.1 Операторы

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
MODAXVAL	F	Определить позицию модулю круговой оси модуло		+	-	PGAsI
MOV	K	Запустить позиционирующую ось		-	+	FBSYsI
MOVt	A	Указать конечную точку движения перемещения в направлении инструмента				FB1(K2)
MSG	P	Программируемые сообщения	M	+	-	PGsI
MVTOOL	P	Языковая команда для движения инструмента		+	-	FBWsI
n	A	Номер вспомогательного кадра ЧПУ		+		PGsI
NAMETOINT	F	Определение индекса системной переменной		+		PGAsI
ЧПУ	K	Спецификация сферы действия данных		+		PGAsI
NEWCONF	P	Применить измененные машинные данные (соответствует "Активировать машинные данные")		+	-	PGAsI
NEWMt	F	Создать новый мультиинструмент		+	-	FBWsI
NEWt	F	Создать новый инструмент		+	-	FBWsI
NORM ⁶⁾	G	Обычная установка в начальной, конечной точке при коррекции инструмента	M	+		PGsI
NOT	K	Логическая НЕ (отрицание)		+		PGAsI
NPROT	P	Специфическая для станка защищенная область ВКЛ/ВЫКЛ		+	-	PGAsI
NPROTDEF	P	Определение специфической для станка защищенной области		+	-	PGAsI
NUMBER	F	Преобразовать входную строку в число		+	-	PGAsI
OEMIPO1	G	Интерполяция OEM 1	M	+		PGAsI
OEMIPO2	G	Интерполяция OEM 2	M	+		PGAsI
OF	K	Кодовое слово в ветвлении CASE		+		PGAsI
OFFN	A	Припуск к запрограммированному контуру	M	+		PGsI
OMA1	A	Адрес OEM 1	M	+		PGAsI
OMA2	A	Адрес OEM 2	M	+		PGAsI
OMA3	A	Адрес OEM 3	M	+		PGAsI
OMA4	A	Адрес OEM 4	M	+		PGAsI
OMA5	A	Адрес OEM 5	M	+		PGAsI
OR	K	Логический оператор, операция ИЛИ		+		PGAsI
ORIXES	G	Линейная интерполяция осей станка или осей ориентации	M	+		PGAsI
ORIXPOS	G	Угол ориентации через виртуальные оси ориентации с позициями круговых осей	M	+		PGAsI
ORIC ⁶⁾	G	Изменения ориентации на наружных углах накладываются на вставляемый круговой кадр	M	+		PGAsI
ORICONCCW	G	Интерполяция на боковой поверхности окружности против часовой стрелки	M	+		PGAsI/FB3sI (F3)

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
ORICONCW	G	Интерполяция на боковой поверхности окружности по часовой стрелке	м	+		PGAsl/FB3sl (F4)
ORICONIO	G	Интерполяция на боковой поверхности окружности с указанием промежуточной ориентации	м	+		PGAsl/FB3sl (F4)
ORICONTO	G	Интерполяция на боковой поверхности окружности в тангенциальном переходе (указание конечной ориентации)	м	+		PGAsl/FB3sl (F5)
ORICURVE	G	Интерполяция ориентации с задачей движения двух контактных точек инструмента	м	+		PGAsl/FB3sl (F6)
ORID	G	Изменения ориентации выполняются перед круговым кадром	м	+		PGAsl
ORIEULER ⁶⁾	G	Угол ориентации через эйлеров угол	м	+		PGAsl
ORIMKS	G	Ориентация инструмента в системе координат станка	м	+		PGAsl
ORIPATH	G	Ориентация инструмента относительно траектории	м	+		PGAsl
ORIPATHS	G	Ориентация инструмента относительно траектории, изгиб в характеристике ориентации сглаживается	м	+		PGAsl
ORIPANE	G	Интерполяция в плоскости (соответствует ORIVECT) Большая круговая интерполяция	м	+		PGAsl
ORIRESET	P	Первичная установка ориентации инструмента с макс. 3 осями ориентации		+	-	PGAsl
ORIROTA ⁶⁾	G	Угол поворота к заданному абсолютно направлению вращения	м	+		PGAsl
ORIROTC	G	Тангенциальный вектор вращения к касательной к траектории	м	+		PGAsl
ORIROTR	G	Угол поворота относительно плоскости между стартовой и конечной ориентацией	м	+		PGAsl
ORIROTT	G	Угол поворота относительно изменения вектора ориентации	м	+		PGAsl
ORIRPY	G	Угол ориентации через угол RPY (XYZ)	м	+		PGAsl
ORIRPY2	G	Угол ориентации через угол RPY (ZYX)	м	+		PGAsl
ORIS	A	Изменение ориентации	м	+		PGAsl
ORISOF ⁶⁾	G	Сглаживание характеристики ориентации ВКЛ	м	+		PGAsl
ORISOLH	F	Вычисление ориентаций		+		PGAsl
ORISON	G	Сглаживание характеристики ориентации ВКЛ	м	+		PGAsl
ORIVECT ⁶⁾	G	Большая круговая интерполяция (идентично ORIPANE)	м	+		PGAsl
ORIVIRT1	G	Угол ориентации через виртуальные оси ориентации (определение 1)	м	+		PGAsl
ORIVIRT2	G	Угол ориентации через виртуальные оси ориентации (определение 1)	м	+		PGAsl

21.1 Операторы

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
ORIWKS ⁶⁾	G	Ориентация инструмента в системе координат детали	м	+		PGAsI
OS	K	Вкл/выкл качания		+		PGAsI
OSB	K	Качание: Начальная точка	м	+		FB1sl (P5)
OSC	G	Постоянное сглаживание ориентации инструмента	м	+		PGAsI
OSCILL	K	AXIS: 1 - 3 оси подачи	м	+		PGAsI
OSCTRL	K	Опции качания	м	+		PGAsI
OSD	G	Перешлифовка ориентации инструмента через задачу длины перешлифовки с SD	м	+		PGAsI
OSE	K	Конечная точка качания	м	+		PGAsI
OSNSC	K	Качание: Количество выхаживаний	м	+		PGAsI
OSOF ⁶⁾	G	Сглаживание ориентации инструмента ВЫКЛ	м	+		PGAsI
OSP1	K	Качание: левая точка возврата	м	+		PGAsI
OSP2	K	Правая точка возврата качания	м	+		PGAsI
OSS	G	Сглаживание ориентации инструмента на конце кадра	м	+		PGAsI
OSSE	G	Сглаживание ориентации инструмента на начале кадра и на конце кадра	м	+		PGAsI
OST	G	Перешлифовка ориентации инструмента через задачу углового допуска в градусах с SD (макс. отклонение от запрограмм. характеристики ориентации)	м	+		PGAsI
OST1	K	Качание: точка останова в левой точке возврата	м	+		PGAsI
OST2	K	Качание: точка останова в правой точке возврата	м	+		PGAsI
OTOL	A	Допуск ориентации для функций компрессора, сглаживания ориентации и типов перешлифовки	м	+		PGAsI
OVR	K	Коррекция частоты вращения	м	+		PGAsI
OVRA	K	Осевая коррекция частоты вращения	м	+	+	PGAsI
OVERRAP	K	Коррекция ускоренного хода	м	+		PGAsI
P	A	Число прогонов подпрограммы		+		PGAsI
PAROT	G	Точная установка системы координат детали на детали	м	+		PGsI
PAROTOF ⁶⁾	G	Отключить относящееся к детали вращение фрейма	м	+		PGsI
PCALL	K	Вызвать подпрограммы с абсолютным указанием пути и передачей параметров		+		PGAsI
PDELAYOF	G	Задержка при штамповке ВЫКЛ	м	+		PGAsI
PDELAYON ⁶⁾	G	Задержка при штамповке ВКЛ	м	+		PGAsI
PHI	K	Угол поворота ориентации вокруг оси направления конуса		+		PGAsI
PHU	K	Физическая единица переменной		+		PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
PL	A	1. В-сплайн: Расстояние между узловыми точками 2. Полиномиальная интерполяция: длина интервала параметров при полиномиальной интерполяции	п	+		PGAsl
PM	K	В минуту		+		PGsl
po	K	Коэффициент полинома при полиномиальной интерполяции	п	+		PGAsl
POCKET3	C (T)	Фрезерование прямоугольного кармана		+		PGAsl
POCKET4	C (T)	Фрезерование кругового кармана		+		PGAsl
POLF	K	Переходная позиция LIFTFAST	м	+		PGsl/PGAsl
POLFA	P	Запустить переходную позицию отдельных осей с \$AA_ESR_TRIGGER	м	+	+	PGsl
POLFMASK	P	Разрешить оси для обратного хода без связи между осями	м	+	-	PGsl
POLFMLIN	P	Разрешить оси для обратного хода с линейной связью между осями	м	+	-	PGsl
POLY	G	полиномиальная интерполяция	м	+		PGAsl
POLYPATH	P	Полиномиальная интерполяция с возможностью выбора для групп осей AXIS или VECT	м	+	-	PGAsl
PON	G	Штамповка ВКЛ	м	+		PGAsl
PONS	G	Штамповка ВКЛ в такте IPO	м	+		PGAsl
Pos	K	Позиционировать ось		+	+	PGsl
POSA	K	Позиционировать ось за границу кадра		+	+	PGsl
POSM	P	Позиционирование магазина		+	-	FBWsl
POSMT	P	Переместить мультиинструмент в зажиме на номер места		+	-	FBWsl
POSP	K	Посегментное позиционирование (качение)		+		PGsl
POSRANGE	F	Определить, находится ли актуальная интерполированная заданная позиция оси в окне вокруг заданной контрольной позиции		+	+	FBSYsl
POT	F	Квадрат (арифметическая функция)		+	+	PGAsl
PR	K	На оборот		+		PGsl
PREPRO	PA	Обозначить подпрограммы с подготовкой		+		PGAsl
PRESETON	P	Установка фактического значения с потерей состояния реферирования		+	+	PGAsl
PRESETONS	P	Установка фактического значения без потери состояния реферирования		+	+	PGAsl
Prio	K	Кодовое слово для установки приоритета при обработке прерываний		+		PGAsl
PRLOC	K	Инициализация переменных при Reset только после локального изменения		+		PGAsl
PROC	K	Первый оператор программы		+		PGAsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
PROTA	P	Запрос повторного расчета модели столкновений		+		PGAsI
PROTD	F	Определение интервала между двумя защищенными областями		+		PGAsI
PROTS	P	Установка состояния защищенной области		+		PGAsI
PSI	K	Апертурный угол конуса		+		PGAsI
PTP	G	Позиционное движение (PTP-перемещение)	м	+		PGAsI
PTPG0	G	Позиционное движение только для G0, в остальных случаях CP	м	+		PGAsI
PTPWOC	G	Позиционное движение без компенсационных движений, вызванных изменениями ориентации	м	+		PGAsI
PUNCHACC	P	Зависящее от пути ускорение при вырубке		+	-	PGAsI
PUTFTOC	P	Точная коррекция инструмента для параллельной правки		+	-	PGAsI
PUTFTOCF	P	Точная коррекция инструмента в зависимости от установленной с FCTDEF функции для параллельной правки		+	-	PGAsI
PW	A	В-сплайн, вес точки	п	+		PGAsI
Qu	K	Быстрый вывод дополнительной (вспомогательной) функции		+		PGsI
R ...	A	R-параметр и как устанавливаемый идентификатор адреса и с числовым расширением		+		PGAsI
RAC	K	Абсолютное покадровое спец. для оси программирование радиуса	п	+		PGsI
RDISABLE	P	Блокировка загрузки		-	+	FBSYsI
READ	P	Считывает в указанном файле одну или несколько строк и сохраняет считанную информацию в поле		+	-	PGAsI
REAL	K	Тип данных: Переменная с плавающей запятой со знаком (вещественные числа)		+		PGAsI
REDEF	K	Переопределение системных переменных, переменных пользователя и языковых команд ЧПУ		+		PGAsI
RELEASE	P	Разрешить оси станка для перехода оси		+	+	PGAsI
REP	K	Кодовое слово для инициализации всех элементов поля с одним и тем же значением		+		PGAsI
REPEAT	K	Повторение программного цикла		+		PGAsI
REPEATB	K	Повторение программной строки		+		PGAsI
REPOSA	G	Повторный подвод к контуру линейный всеми осями	п	+		PGAsI
REPOSH	G	Повторный подвод к контуру по полукругу	п	+		PGAsI
REPOSHA	G	Повторный подвод к контуру всеми осями; геометрические оси по полукругу	п	+		PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
REPOSL	G	Повторный подвод к контуру линейный	п	+		PGAsl
REPOSQ	G	Повторный подвод к контуру по четверти круга	п	+		PGAsl
REPOSQA	G	Повторный подвод к контуру линейно всеми осями; геометрические оси по четверти круга	п	+		PGAsl
RESETMON	P	Языковая команда для активации заданного значения		+	-	FBWsl
RET	P	Конец подпрограммы		+	+	PGAsl
RETB	P	Конец подпрограммы		+	+	PGAsl
RIC	K	Относительное покадровое спец. для оси программирование радиуса	п	+		PGsl
RINDEX	F	Определить индекс символа во входной строке		+	-	PGAsl
RMB	G	Повторный подвод к начальной точке кадра	м	+		PGAsl
RMBBL	G	Повторный подвод к начальной точке кадра	п	+		PGAsl
RME	G	Повторный подвод к конечной точке кадра	м	+		PGAsl
RMEBL	G	Повторный подвод к конечной точке кадра	п	+		PGAsl
RMI ⁶⁾	G	Повторный подвод к точке прерывания	м	+		PGAsl
RMIBL ⁶⁾	G	Повторный подвод к точке прерывания	п	+		PGAsl
RMN	G	Повторный подвод к следующей точке траектории	м	+		PGAsl
RMNBL	G	Повторный подвод к следующей точке траектории	п	+		PGAsl
RND	A	Закругление угла контура	п	+		PGsl
RNDM	A	Модальное закругление	м	+		PGsl
ROT	G	Программируемое вращение	п	+		PGsl
ROTS	G	Программируемые вращения фрейма с пространственными углами	п	+		PGsl
ROUND	F	Округление мест после запятой		+	+	PGAsl
ROUNDUP	F	Округление входного значения в сторону увеличения		+	+	PGAsl
RP	A	Полярный радиус	м/п	+		PGsl
RPL	A	Вращение в плоскости	п	+		PGsl
RT	K	Параметры для доступа к данным фрейма: Вращение		+		PGAsl
RTLIOF	G	G0 без линейной интерполяции (интерполяция отдельной оси)	м	+		PGsl
RTLION ⁶⁾	G	G0 с линейной интерполяцией	м	+		PGsl

Операторы S ... Z

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
S	A	Частота вращения шпинделя (для G4, G96/G961 другое значение)	м/п	+	+	PGsl
Save	PA	Атрибут для сохранения информации при вызовах подпрограмм		+		PGAsl
SBLOF	P	Подавление отдельного кадра		+	-	PGAsl
SBLON	P	Отменить подавление отдельного кадра		+	-	PGAsl
sc	K	Параметры для доступа к данным фрейма: Масштабирование		+		PGAsl
SCALE	G	Программируемое масштабирование	п	+		PGsl
SCC	K	Выборочное согласование поперечной оси с G96/G961/G962. Идентификаторами оси могут быть гео-ось, ось канала или ось станка.		+		PGsl
SCPARA	K	Программирование блока параметров Servo		+	+	PGAsl
SD	A	Порядок слайна	п	+		PGAsl
set	K	Кодовое слово для инициализации всех элементов поля с перечисленными значениями		+		PGAsl
SETAL	P	Установить аварийное сообщение		+	+	PGAsl
SETDNO	F	Согласовать номер D резца (CE) инструмента (T)		+	-	PGAsl
SETINT	K	Определение, какой обработчик прерываний должен быть активирован, если вход ЧПУ готов		+		PGAsl
SETM	P	Установка маркеров в собственном канале		+	+	PGAsl
SETMS	P	Переключение на установленный в машинных данных мастер-шпиндель		+	-	PGsl
SETMS(n)	P	Шпиндель n должен считаться мастер-шпинделем		+		PGsl
SETMTH	P	Установить номер ведущего INSTR. суппорта		+	-	FBWsl
SETPIECE	P	Учитывать количество штук для всех инструментов, которые согласованы со шпинделем.		+	-	FBWsl
SETTA	P	Активация инструмента из структуры износа		+	-	FBWsl
SETTCOR	F	Изменение компонентов инструмента с учетом всех граничных условий		+	-	PGAsl
SETTIA	P	Деактивация инструмента из структуры износа		+	-	FBWsl
SF	A	Смещение стартовой точки для резьбонарезания	м	+		PGsl
SIN	F	Синус (тригонометрическая функция)		+	+	PGAsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
SIRELAY	F	Активировать спараметрированные со SIRELIN, SIRELOUT и SIRELTIME функции безопасности		-	+	FBSIsI
SIRELIN	P	Инициализировать входные величины функционального блока		+	-	FBSIsI
SIRELOUT	P	Инициализировать выходные величины функционального блока		+	-	FBSIsI
SIRELTIME	P	Инициализировать таймер функционального блока		+	-	FBSIsI
SLOT1	C (T)	Продольный паз		+		PGAsI
SLOT2	C (T)	Кольцевая канавка		+		PGAsI
SOFT	G	Ускорение по траектории с ограничением рывка	м	+		PGAsI
SOFTA	P	Включение осевого ускорения с ограничением рывка для запрограммированных осей		+	-	PGAsI
SON	G	Вырубка ВКЛ	м	+		PGAsI
SONS	G	Вырубка ВКЛ в такте IPO	м	+		PGAsI
SPATH ⁶⁾	G	Соотношение траекторий для осей FGROUP это длина дуги	м	+		PGAsI
SPCOF	P	Переключение мастер-шпинделя или шпинделя(ей) из режима управления по положению в режим управления по частоте вращения	м	+	-	PGsI
SPCON	P	Переключение мастер-шпинделя или шпинделя(ей) из режима управления по частоте вращения в режим управления по положению	м	+	-	PGAsI
SPI	F	Преобразует номер шпинделя в идентификатор оси		+	-	PGAsI
SPIF1 ⁶⁾	G	Быстрые входы/выходы ЧПУ для штамповки/вырубки Байт 1	м	+		FB2sI (N4)
SPIF2	G	Быстрые входы/выходы ЧПУ для штамповки/вырубки Байт 2	м	+		FB2sI (N4)
SPLINEPATH	P	Определение соединения сплайнов		+	-	PGAsI
SPN	A	Кол-во участков пути на кадр	п	+		PGAsI
SPOF ⁶⁾	G	Ход ВЫКЛ, штамповка, вырубка ВЫКЛ	м	+		PGAsI
SPOS	K	Позиция шпинделя	м	+	+	PGsI
SPOSA	K	Позиция шпинделя за границы кадра	м	+		PGsI
SPP	A	Длина участка пути	м	+		PGAsI
SPRINT	F	Возвращает отформатированную входную строку		+		PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
SQRT	F	Квадратный корень (арифметическая функция) (square root)		+	+	PGAsI
SR	A	Путь обратного хода качания для синхронного действия	п	+		PGsI
SRA	K	Путь обратного хода качания при внешнем входе осевом для синхронного действия	м	+		PGsI
ST	A	Время выхаживания качания для синхронного действия	п	+		PGsI
STA	K	Время выхаживания качания осевого для синхронного действия	м	+		PGsI
START	P	Запуск выбранных программ в нескольких каналах одновременно из текущей программы		+	-	PGAsI
STARTFIFO ⁶⁾	G	Выполнение; параллельное заполнение буфера предварительной обработки	м	+		PGAsI
STAT		Положение шарниров	п	+		PGAsI
STOLF	K	Коэффициент допуска G0	м	+		PGAsI
STOPFIFO	G	Остановка обработки; заполнение буфера предварительной обработки, до распознавания STARTFIFO, пока буфер предварительной обработки не будет заполнен или до конца программы	м	+		PGAsI
stopre	P	Остановка предварительной обработки до выполнения всех подготовленных кадров главного хода		+	-	PGAsI
STOPREOF	P	Отменить остановку предварительной обработки		-	+	FBSYsI
String	K	Тип данных: строка символов		+		PGAsI
STRINGIS	F	Проверяет имеющуюся языковую среду ЧПУ и специально относящиеся к этой команде имена циклов ЧПУ, переменные пользователя, макросы и имена меток, существуют ли они, определены ли они или активны.		+	-	PGAsI
STRLEN	F	Определить длину строки		+	-	PGAsI
SUBSTR	F	Определить индекс символа во входной строке		+	-	PGAsI
SUPA	G	Подавление актуального смещения нулевой точки, вкл. запрограммированные смещения, системные фреймы, смещения маховичком (DRF), внешнее смещение нулевой точки и наложенное движение	п	+		PGsI
SVC	K	Скорость резания инструмента	м	+		PGsI
SYNFCT	P	Обработка полинома в зависимости от условия в синхронном действии движения		-	+	FBSYsI
SYNR	K	Чтение переменной выполняется синхронного, т.е. на момент времени выполнения		+		PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
SYNRW	K	Чтение и запись переменной выполняется синхронного, т.е. на момент времени выполнения		+		PGAsI
SYNW	K	Запись переменной выполняется синхронного, т.е. на момент времени выполнения		+		PGAsI
T	A	Вызвать инструмент (смена только если установлено в машинных данных; в ином случае необходима команда M6)		+		PGsI
TAN	F	Тангенс (тригонометрическая функция)		+	+	PGAsI
TANG	P	Тангенциальное управление: Определить соединение		+	-	PGAsI
TANGDEL	P	Тангенциальное управление: Удаление соединения		+	-	PGAsI
TANGOF	P	Тангенциальное управление: Выключение соединения		+	-	PGAsI
TANGON	P	Тангенциальное управление: Включить соединение		+	-	PGAsI
TCA (828D: _TCA)	P	Выбор инструмента / смена инструмента независимо от состояния инструмента		+	-	FBWsl
TCARR	A	Запросить инструментальный суппорт (номер "m")		+		PGAsI
TCI	P	Поместить инструмент из буфера в магазин		+	-	FBWsl
TCOABS ⁶⁾	G	Определение компонентов длин инструмента из актуальной ориентации инструмента	м	+		PGAsI
TCOFR	G	Определить компоненты длин инструмента из ориентации активного фрейма	м	+		PGAsI
TCOFRX	G	Определение ориентации инструмента активного фрейма при выборе инструмента, инструмент показывает в направлении X	м	+		PGAsI
TCOFRY	G	Определение ориентации инструмента активного фрейма при выборе инструмента, инструмент показывает в направлении Y	м	+		PGAsI
TCOFRZ	G	Определение ориентации инструмента активного фрейма при выборе инструмента, инструмент показывает в направлении Z	м	+		PGAsI
THETA	A	Угол поворота	п	+		PGAsI
TILT	A	Боковой угол	м	+		PGAsI
TLIFT	P	Тангенциальное управление: Включить создание промежуточных кадров		+	-	PGAsI
TML	P	Выбор инструмента с номером места в магазине		+	-	FBWsl
TMOF	P	Отменить контроль инструмента		+	-	PGAsI
TMON	P	Выбрать контроль инструмента		+	-	PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
TO	K	Обозначает конечное значение в цикле вычисления FOR		+		PGAsI
TOFF	K	Смещение длин инструмента в направлении компонента длин инструмента, действующего параллельно указанной в индексе гео-оси.	M	+		PGsI
TOFFL	K	Смещение длин инструмента в направлении компонента длин инструмента L1, L2 или L3	M	+		PGsI
TOFFOF	P	Сбросить онлайн-коррекцию длины инструмента		+	-	PGAsI
TOFFON	P	Активировать онлайн-коррекцию длины инструмента		+	-	PGAsI
TOFFR	A	Смещение радиуса инструмента	M	+		PGsI
TOFRAME	G	Точно установить ось Z СКД через вращение фрейма параллельно ориентации инструмента	M	+		PGsI
TOFRAMEX	G	Точно установить ось X СКД через вращение фрейма параллельно ориентации инструмента	M	+		PGsI
TOFRAMEY	G	Точно установить ось Y СКД через вращение фрейма параллельно ориентации инструмента	M	+		PGsI
TOFRAMEZ	G	как TOFRAME	M	+		PGsI
TOLOWER	F	Преобразовать буквы строки в строчные буквы		+	-	PGAsI
TOOLENV	F	Сохранить все состояния, важные для нормирования находящихся в памяти параметров инструмента		+	-	PGAsI
TOOLGNT	F	Определить число инструментов одной группы инструментов		+	-	FBWsI
TOOLGT	F	Определить T-номер инструмента из группы инструментов		+	-	FBWsI
TOROT	G	Точно установить ось Z СКД через вращение фрейма параллельно ориентации инструмента	M	+		PGsI
TOROTOF ⁶⁾	G	Вращение фрейма в направлении инструмента ВЫКЛ	M	+		PGsI
TOROTX	G	Точно установить ось X СКД через вращение фрейма параллельно ориентации инструмента	M	+		PGsI
TOROTY	G	Точно установить ось Y СКД через вращение фрейма параллельно ориентации инструмента	M	+		PGsI
TOROTZ	G	как TOROT	M	+		PGsI
TOUPPER	F	Преобразовать буквы строки в прописные буквы		+	-	PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
TOWBCS	G	Значения износа в базовой кинематической системе (BKS)	м	+		PGAsl
TOWKCS	G	Значения износа в системе координат инструментальной головки при кинематической трансформации (отличается от MCS вращением инструмента)	м	+		PGAsl
TOWMCS	G	Значения износа в системе координат станка (MCS)	м	+		PGAsl
TOWSTD ⁶⁾	G	Значение по умолчанию для коррекции по длине инструмента	м	+		PGAsl
TOWTCS	G	Значения износа в системе координат инструмента (исходная точка инстр. суппорта T на креплении суппорта)	м	+		PGAsl
TOWWCS	G	Значения износа в системе координат детали (WCS)	м	+		PGAsl
tr	K	Компонент смещения переменной фрейма		+		PGAsl
TRAANG	P	Трансформация наклонной оси		+	-	PGAsl
TRACON	P	Каскадированная трансформация		+	-	PGAsl
TRACYL	P	Цилиндр: трансформация боковой поверхности		+	-	PGAsl
TRAFOOF	P	Отключить активные в канале трансформации		+	-	PGAsl
TRAFOON	P	Включение трансформации, определенной посредством кинематических цепочек		+	-	PGAsl
TRAILOF	P	Осевая синхронная буксировка ВЫКЛ		+	+	PGAsl
TRAILON	P	Осевая синхронная буксировка ВКЛ		+	+	PGAsl
TRANS	G	Абсолютное программируемое смещение нулевой точки	п	+		PGsl
Transmit	P	Полярная трансформация (обработка торцовых поверхностей)		+	-	PGAsl
TRAORI	P	4-, 5-осевая трансформация, базовая трансформация		+	-	PGAsl
true	K	Логическая постоянная: истинно		+		PGAsl
TRUNC	F	Отсечение мест после запятой		+	+	PGAsl
Tu		Осевой угол	п	+		PGAsl
Turn	A	Число витков для спиральной линии	п	+		PGsl
ULI	K	Верхнее предельное значение переменной		+		PGAsl
UNLOCK	P	Разрешить синхронное действие с ID (продолжить технологический цикл)		-	+	FBSYsl
UNTIL	K	Условие для завершения цикла REPEAT		+		PGAsl
UPATH	G	Соотношение траекторий для осей FGROUP это параметр кривой	м	+		PGAsl
Var	K	Кодовое слово: тип передачи параметров		+		PGAsl
VELOLIM	K	Уменьшение макс. осевой скорости	м	+		PGAsl
VELOLIMA	K	Уменьшение или увеличение макс. осевой скорости ведомой оси	м	+	+	PGAsl

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
1) 2) 3) 4) 5) Объяснения см. Экспликация (Страница 914).						
WAITC	P	Ожидание выполнения критерия смены кадра соединения для оси/шпинделя		+	-	PGAsI
WAITE	P	Ожидание конца программы в другом канале.		+	-	PGAsI
WAITENC	P	Ожидание синхронизированных или восстановленных позиций осей		+	-	PGAsI
WAITM	P	Ожидание метки в указанном канале; завершить предшествующий кадр с точным остановом.		+	-	PGAsI
WAITMC	P	Ожидание метки в указанном канале; точный останов только если другие каналы еще не достигли метки		+	-	PGAsI
WAITP	P	Ожидание конца перемещения позиционирующей оси		+	-	PGsI
WAITS	P	Ожидание достижения позиции шпинделя		+	-	PGsI
WALCS0 ⁶⁾	G	Ограничение рабочего поля WCS отключено	м	+	-	PGsI
WALCS1	G	Группа 1 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALCS2	G	Группа 2 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALCS3	G	Группа 3 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALCS4	G	Группа 4 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALCS5	G	Группа 5 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALCS6	G	Группа 6 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALCS7	G	Группа 7 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALCS8	G	Группа 8 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALCS9	G	Группа 9 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALCS10	G	Группа 10 ограничений рабочего поля СКД активна	м	+	-	PGsI
WALIMOF	G	Ограничение рабочего поля BKS ВЫКЛ	м	+	-	PGsI
WALIMON ⁶⁾	G	Ограничение рабочего поля BKS ВКЛ	м	+	-	PGsI
WHEN	K	Операция будет выполнена один раз, если условие выполнено один раз.		-	+	FBSYsI
WHENEVER	K	Операция выполняется циклически в каждом такте интерполяции, если условие выполнено.		-	+	FBSYsI
WHILE	K	Начало программного цикла WHILE		+		PGAsI
write	P	Записать текст в файловую систему. Добавляет кадр в конце указанного файла.		+	-	PGAsI

Оператор	Тип ¹⁾	Значение	W ²⁾	TP ³⁾	SA ⁴⁾	Описание см. ⁵⁾
^{1) 2) 3) 4) 5)} Объяснения см. Экспликация (Страница 914).						
WRTPR	P	Запись строки символов в переменную BTSS		+	-	PGsl
X	A	Имя оси	м/п	+	+	PGsl
XOR	O	Логическая исключаяющая ИЛИ		+		PGAsl
Y	A	Имя оси	м/п	+	+	PGsl
Z	A	Имя оси	м/п	+	+	PGsl

Легенда

1) Тип оператора:

A Адрес

Идентификатор, которому присваивается значение (к примеру, OVR=10). Также есть несколько адресов, которые включают или выключают функцию без присвоения значения (к примеру, CPLON и CPLOF).

C (A) Цикл AST

Предопределенная программа ЧПУ для дополнительной автоматической оптимизации с AST (= автоматическая оптимизация Servo) Согласование с конкретной ситуацией оптимизации осуществляется по параметрам, которые передаются циклу при вызове.

C (M) Цикл измерения

Предопределенная программа ЧПУ, в которой универсально запрограммирован определенный процесс измерения, например, определение внутреннего диаметра цилиндрической детали. Согласование с конкретной ситуацией измерения осуществляется по параметрам, которые передаются циклу при вызове.

C (T) Технологический цикл

Предопределенная программа ЧПУ, в которой универсально запрограммирован определенный процесс обработки, например, нарезание внутренней резьбы или фрезерование кармана. Адаптация к конкретной ситуации обработки осуществляется через параметры, которые передаются циклу при вызове.

F Предопределенная функция (выводит возвращаемое значение)

Вызов предопределенной функции может быть указан как операнд в выражении.

G G-команда

G-команды разделены на группы. В одном кадре можно записать только одну G-команду группы. G-команда может действовать модально (до отмены через другую команду этой же группы), или она действует только для кадра, в котором она стоит (действует покадрово).

K Кодовое слово

Идентификатор, определяющий синтаксис кадра. Кодовому слову значение не присваивается и включение/выключение функции ЧПУ с помощью кодового слова невозможно.

Примеры: Управляющие структуры (IF, ELSE, ENDIF, WHEN, ...), выполнение программы (GOTO, GOTO, RET ...)

O Оператор

Оператора для математической операции, операции сравнения или логической операции

P Предопределенная процедура (не выводит возвращаемое значение)

PA Атрибут программы

Атрибуты программы стоят в конце строки определения подпрограммы:

PROC <имя программы>(...) <атрибут программы>

Они определяют поведение при выполнении подпрограммы.

2) Активность оператора:

m модально

p покадрово

- 3) Возможность программирования в программе обработки детали:
- + программируемые
 - не программируемые
- M программируется только изготовителем станка
- 4) Возможность программирования в синхронных действиях:
- + программируемые
 - не программируемые
- T программируется только в технологических циклах
- 5) Ссылка на документ, содержащий подробное описание оператора:
- | | |
|------------------|--|
| <i>PGsl</i> | Руководство по программированию "Основы" |
| <i>PGAsl</i> | Руководство по программированию "Расширенное программирование" |
| <i>BNMsl</i> | Руководство по программированию "Измерительные циклы" |
| <i>BHDsl</i> | Руководство оператора "Токарная обработка" |
| <i>BHFsl</i> | Руководство оператора "Фрезерная обработка" |
| <i>FB1sl ()</i> | Описание функций "Основные функции" (с алфавитно-цифровым сокращением соответствующего описания функций в скобках) |
| <i>FB2sl ()</i> | Описание функций "Дополнительные функции" (с алфавитно-цифровым сокращением соответствующего описания функций в скобках) |
| <i>FB3sl ()</i> | Описание функций "Специальные функции" (с алфавитно-цифровым сокращением соответствующего описания функций в скобках) |
| <i>FBSsl</i> | Описание функций Safety Integrated |
| <i>FBSYsl</i> | Описание функций "Синхронные действия" |
| <i>FBWsl</i> | Описание функций "Управление инструментом" |
- 6) Стандартная установка в начале программы (в состоянии СЧПУ при поставке, когда еще ничего другого не запрограммировано).

Рисунок 21-1 Значение сносок в таблицах операторов

21.2 Операторы: Доступность для SINUMERIK 828D

Примечание

Циклы

Циклы обозначены как "опциональные", если они зависят от следующих опций с обязательным лицензированием:

- Расширенные технологические функции (номер артикула: 6FC5800-0AP58-0YB0)
- Измерительные циклы (номер артикула: 6FC5800-0AP28-0YB0)
- Измерение кинематики (номер артикула: 6FC5800-0AP18-0YB0)
- SINUMERIK: шлифование (расширенная версия Advanced (номер артикула: 6FC5800-0AS35-0YB0)

Обозначения нет, если циклы из-за опции "Расширенные технологические функции" имеют в определенных системах частичный функционал.

21.2.1 Варианты управления: фрезерование / токарная обработка

Операторы A ... C

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (mε42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (tε42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (mε62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (tε62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (mε821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (mε822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (tε821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (tε822)
:	•	•	•	•	•	•	•	•
*	•	•	•	•	•	•	•	•
+	•	•	•	•	•	•	•	•
-	•	•	•	•	•	•	•	•
<	•	•	•	•	•	•	•	•
<<	•	•	•	•	•	•	•	•
<=	•	•	•	•	•	•	•	•
=	•	•	•	•	•	•	•	•
>=	•	•	•	•	•	•	•	•
/	•	•	•	•	•	•	•	•
/0 ... /7	•	•	•	•	•	•	•	•
A	•	•	•	•	•	•	•	•
A2	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-
A4	-	-	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
A5	-	-	-	-	-	-	-	-
A6	-	-	-	-	-	-	-	-
A7	-	-	-	-	-	-	-	-
ABS	•	•	•	•	•	•	•	•
AC	•	•	•	•	•	•	•	•
ACC	•	•	•	•	•	•	•	•
ACCLIMA	•	•	•	•	•	•	•	•
ACN	•	•	•	•	•	•	•	•
ACOS	•	•	•	•	•	•	•	•
ACP	•	•	•	•	•	•	•	•
ACTBLOCNO	•	•	•	•	•	•	•	•
ADDFRAME	•	•	•	•	•	•	•	•
ADIS	•	•	•	•	•	•	•	•
ADISPOS	•	•	•	•	•	•	•	•
ADISPOSA	•	•	•	•	•	•	•	•
ALF	•	•	•	•	•	•	•	•
AMIRROR	•	•	•	•	•	•	•	•
AND	•	•	•	•	•	•	•	•
ANG	•	•	•	•	•	•	•	•
AP	•	•	•	•	•	•	•	•
APR	•	•	•	•	•	•	•	•
APRB	•	•	•	•	•	•	•	•
APRP	•	•	•	•	•	•	•	•
APW	•	•	•	•	•	•	•	•
APWB	•	•	•	•	•	•	•	•
APWP	•	•	•	•	•	•	•	•
APX	•	•	•	•	•	•	•	•
AR	•	•	•	•	•	•	•	•
AROT	•	•	•	•	•	•	•	•
AROTS	•	•	•	•	•	•	•	•
AS	•	•	•	•	•	•	•	•
ASCALE	•	•	•	•	•	•	•	•
ASIN	•	•	•	•	•	•	•	•
ASPLINE	○	○	○	○	○	○	○	○
ATAN2	•	•	•	•	•	•	•	•
ATOL	•	•	•	•	•	•	•	•
ATRANS	•	•	•	•	•	•	•	•

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
AUXFUDEL	•	•	•	•	•	•	•	•
AUXFUDELG	•	•	•	•	•	•	•	•
AUXFUMSEQ	•	•	•	•	•	•	•	•
AUXFUSYNC	•	•	•	•	•	•	•	•
AX	•	•	•	•	•	•	•	•
AXCTSWE	-	-	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-	-	-
AXIS	•	•	•	•	•	•	•	•
AXNAME	•	•	•	•	•	•	•	•
AXSTRING	•	•	•	•	•	•	•	•
AXTOCHAN	•	•	•	•	•	•	•	•
AXTOSPI	•	•	•	•	•	•	•	•
B	•	•	•	•	•	•	•	•
B2	-	-	-	-	-	-	-	-
B3	-	-	-	-	-	-	-	-
B4	-	-	-	-	-	-	-	-
B5	-	-	-	-	-	-	-	-
B6	-	-	-	-	-	-	-	-
B7	-	-	-	-	-	-	-	-
B_AND	•	•	•	•	•	•	•	•
B_OR	•	•	•	•	•	•	•	•
B_NOT	•	•	•	•	•	•	•	•
B_XOR	•	•	•	•	•	•	•	•
BAUTO	○	○	○	○	○	○	○	○
BLOCK	•	•	•	•	•	•	•	•
BLSYNC	•	•	•	•	•	•	•	•
BNAT	○	○	○	○	○	○	○	○
BOOL	•	•	•	•	•	•	•	•
BOUND	•	•	•	•	•	•	•	•
BRISK	•	•	•	•	•	•	•	•
BRISKA	•	•	•	•	•	•	•	•
BSPLINE	○	○	○	○	○	○	○	○
BTAN	○	○	○	○	○	○	○	○
C	•	•	•	•	•	•	•	•
C2	-	-	-	-	-	-	Имя оси канала	Имя оси ка- нала

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
C3	-	-	-	-	-	-	-	-
C4	-	-	-	-	-	-	-	-
C5	-	-	-	-	-	-	-	-
C6	-	-	-	-	-	-	-	-
C7	-	-	-	-	-	-	-	-
CAC	•	•	•	•	•	•	•	•
CACN	•	•	•	•	•	•	•	•
CACP	•	•	•	•	•	•	•	•
CALCDAT	•	•	•	•	•	•	•	•
CALCPOSI	•	•	•	•	•	•	•	•
CALL	•	•	•	•	•	•	•	•
CALLPATH	•	•	•	•	•	•	•	•
CANCEL	•	•	•	•	•	•	•	•
CASE	•	•	•	•	•	•	•	•
CDC	•	•	•	•	•	•	•	•
CDOF	-	-	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-	-	-
CDON	-	-	-	-	-	-	-	-
CFC	•	•	•	•	•	•	•	•
CFIN	•	•	•	•	•	•	•	•
CFINE	•	•	•	•	•	•	•	•
CFTCP	•	•	•	•	•	•	•	•
Chan	•	•	•	•	•	•	•	•
CHANDATA	•	•	•	•	•	•	•	•
CHAR	•	•	•	•	•	•	•	•
CHF	•	•	•	•	•	•	•	•
CHKDM	•	•	•	•	•	•	•	•
CHKDNO	•	•	•	•	•	•	•	•
CHR	•	•	•	•	•	•	•	•
CIC	•	•	•	•	•	•	•	•
CIP	•	•	•	•	•	•	•	•
CLEARM	-	-	-	-	-	○	-	•
CLRINT	•	•	•	•	•	•	•	•
CMIRROR	•	•	•	•	•	•	•	•
COARSEA	•	•	•	•	•	•	•	•
COLLPAIR	-	-	-	-	-	-	-	-
COMPCAD	•	-	•	-	•	•	-	-

Оператор	SINUMERIK 828D							
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
COMPCURV	●	-	●	-	●	●	-	-
COMPLETE	●	●	●	●	●	●	●	●
COMPOF	●	-	●	-	●	●	-	-
COMPON	●	-	●	-	●	●	-	-
COMPSURF	-	-	○	-	○	○	-	-
CONTDCON	●	●	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●	●	●
CORROF	●	●	●	●	●	●	●	●
CORRTRAF0	-	-	-	-	-	-	-	-
COS	●	●	●	●	●	●	●	●
COUPDEF	-	○	-	○	-	-	○	○
COUPDEL	-	○	-	○	-	-	○	○
COUPOF	-	○	-	○	-	-	○	○
COUPOFS	-	○	-	○	-	-	○	○
COUPON	-	○	-	○	-	-	○	○
COUPONC	-	○	-	○	-	-	○	○
COUPRES	-	○	-	○	-	-	○	○
CP	●	●	●	●	●	●	●	●
CPBC	○	○	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○	○	○
CPFM0F	○	○	○	○	○	○	○	○
CPFM0N	○	○	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○	○	○
CPFRS	○	○	○	○	○	○	○	○
CPLA	○	○	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○	○	○
CPLINTR	○	○	○	○	○	○	○	○
CPLNUM	○	○	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○	○	○
CPLON	○	○	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○	○	○

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
CPLOUTTR	○	○	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○	○	○
CPOF	○	○	○	○	○	○	○	○
CPON	○	○	○	○	○	○	○	○
CPRECOF	•	•	•	•	•	•	•	•
CPRECON	•	•	•	•	•	•	•	•
CPRES	○	○	○	○	○	○	○	○
CPROT	•	•	•	•	•	•	•	•
CPROTDEF	•	•	•	•	•	•	•	•
CPSETTYPE	○	○	○	○	○	○	○	○
CPSYNCOF	○	○	○	○	○	○	○	○
CPSYNCOF2	○	○	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○	○	○
CR	•	•	•	•	•	•	•	•
CROT	•	•	•	•	•	•	•	•
CROTS	•	•	•	•	•	•	•	•
CRPL	•	•	•	•	•	•	•	•
CSCALE	•	•	•	•	•	•	•	•
CSPLINE	○	○	○	○	○	○	○	○
CT	•	•	•	•	•	•	•	•
CTAB	-	-	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-	-	-

Оператор	SINUMERIK 828D							
	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
CTABFSEG	-	-	-	-	-	-	-	-
CTABID	-	-	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-	-	-
CTABNO	-	-	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-	-	-
CTABPOLID	-	-	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-	-	-
CTABSEGID	-	-	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-	-	-
CTOL	•	•	•	•	•	•	•	•
CTRANS	•	•	•	•	•	•	•	•
CUT2D	•	•	•	•	•	•	•	•
CUT2DD	•	•	•	•	•	•	•	•
CUT2DF	•	•	•	•	•	•	•	•
CUT2DFD	•	•	•	•	•	•	•	•
CUT3DC	-	-	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
CUT3DFS	-	-	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●	●	●
CUTCONON	●	●	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●	●	●
CUTMODK	-	-	-	-	-	-	-	-
CYCLE60	○	○	●	●	●	●	●	●
CYCLE61	●	●	●	●	●	●	●	●
CYCLE62	●	●	●	●	●	●	●	●
CYCLE63	○	○	●	●	●	●	●	●
CYCLE64	○	○	●	●	●	●	●	●
CYCLE70	○	○	●	●	●	●	●	●
CYCLE72	●	●	●	●	●	●	●	●
CYCLE76	●	●	●	●	●	●	●	●
CYCLE77	●	●	●	●	●	●	●	●
CYCLE78	○	○	●	●	●	●	●	●
CYCLE79	○	○	●	●	●	●	●	●
CYCLE81	●	●	●	●	●	●	●	●
CYCLE82	●	●	●	●	●	●	●	●
CYCLE83	●	●	●	●	●	●	●	●
CYCLE84	●	●	●	●	●	●	●	●
CYCLE85	●	●	●	●	●	●	●	●
CYCLE86	●	●	●	●	●	●	●	●
CYCLE92	●	●	●	●	●	●	●	●
CYCLE95	●	●	●	●	●	●	●	●
CYCLE98	●	●	●	●	●	●	●	●
CYCLE99	●	●	●	●	●	●	●	●
CYCLE150	○	○	○	○	○	○	○	○
CYCLE435	-	-	-	-	-	-	-	-
CYCLE495	-	-	-	-	-	-	-	-
CYCLE750	●	●	●	●	●	●	●	●
CYCLE751	●	●	●	●	●	●	●	●
CYCLE752	●	●	●	●	●	●	●	●
CYCLE753	●	●	●	●	●	●	●	●
CYCLE754	●	●	●	●	●	●	●	●
CYCLE755	●	●	●	●	●	●	●	●
CYCLE756	●	●	●	●	●	●	●	●
CYCLE757	●	●	●	●	●	●	●	●

Оператор	SINUMERIK 828D							
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
CYCLE758	●	●	●	●	●	●	●	●
CYCLE759	●	●	●	●	●	●	●	●
CYCLE800	-	-	●	●	●	●	●	●
CYCLE801	●	●	●	●	●	●	●	●
CYCLE802	●	●	●	●	●	●	●	●
CYCLE830	○	○	●	●	●	●	●	●
CYCLE832	●	●	●	●	●	●	●	●
CYCLE840	●	●	●	●	●	●	●	●
CYCLE899	○	○	●	●	●	●	●	●
CYCLE930	●	●	●	●	●	●	●	●
CYCLE940	●	●	●	●	●	●	●	●
CYCLE951	●	●	●	●	●	●	●	●
CYCLE952	○	○	●	●	●	●	●	●
CYCLE961	○	○	○	○	○	○	○	○
CYCLE971	○	○	○	○	○	○	○	○
CYCLE973	○	○	○	○	○	○	○	○
CYCLE974	○	○	○	○	○	○	○	○
CYCLE976	○	○	○	○	○	○	○	○
CYCLE977	○	○	○	○	○	○	○	○
CYCLE978	○	○	○	○	○	○	○	○
CYCLE979	○	○	○	○	○	○	○	○
CYCLE982	○	○	○	○	○	○	○	○
CYCLE994	○	○	○	○	○	○	○	○
CYCLE995	○	○	○	○	○	○	○	○
CYCLE996	○	-	○	-	○	○	-	-
CYCLE997	○	○	○	○	○	○	○	○
CYCLE998	○	○	○	○	○	○	○	○
CYCLE4071	-	-	-	-	-	-	-	-
CYCLE4072	-	-	-	-	-	-	-	-
CYCLE4073	-	-	-	-	-	-	-	-
CYCLE4074	-	-	-	-	-	-	-	-
CYCLE4075	-	-	-	-	-	-	-	-
CYCLE4077	-	-	-	-	-	-	-	-
CYCLE4078	-	-	-	-	-	-	-	-
CYCLE4079	-	-	-	-	-	-	-	-

Операторы D ... F

Оператор	SINUMERIK 828D							
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
D	●	●	●	●	●	●	●	●
D0	●	●	●	●	●	●	●	●
DAC	●	●	●	●	●	●	●	●
DC	●	●	●	●	●	●	●	●
DCI	●	●	●	●	●	●	●	●
DCM	●	●	●	●	●	●	●	●
DCU	●	●	●	●	●	●	●	●
DEF	●	●	●	●	●	●	●	●
DEFINE	●	●	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●	●	●
DELDL	●	●	●	●	●	●	●	●
DELDTG	●	●	●	●	●	●	●	●
DELETE	●	●	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●	●	●
DEMLRES	●	●	●	●	●	●	●	●
DELMT	●	●	●	●	●	●	●	●
DELOBJ	-	-	-	-	-	-	-	-
DELT	●	●	●	●	●	●	●	●
DELTC	●	●	●	●	●	●	●	●
DELTOOLNV	●	●	●	●	●	●	●	●
DIACYCOFA	●	●	●	●	●	●	●	●
DIAM90	●	●	●	●	●	●	●	●
DIAM90A	●	●	●	●	●	●	●	●
DIAMCHAN	●	●	●	●	●	●	●	●
DIAMCHANA	●	●	●	●	●	●	●	●
DIAMCYCOF	●	●	●	●	●	●	●	●
DIAMOF	●	●	●	●	●	●	●	●
DIAMOFA	●	●	●	●	●	●	●	●
DIAMON	●	●	●	●	●	●	●	●
DIAMONA	●	●	●	●	●	●	●	●
DIC	●	●	●	●	●	●	●	●
DILF	●	●	●	●	●	●	●	●
Disable	●	●	●	●	●	●	●	●

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
DISC	•	•	•	•	•	•	•	•
DISCL	•	•	•	•	•	•	•	•
DISPLOF	•	•	•	•	•	•	•	•
DISPLON	•	•	•	•	•	•	•	•
DISPR	•	•	•	•	•	•	•	•
DISR	•	•	•	•	•	•	•	•
DISRP	•	•	•	•	•	•	•	•
DITE	•	•	•	•	•	•	•	•
DITS	•	•	•	•	•	•	•	•
DIV	•	•	•	•	•	•	•	•
dl	-	-	-	-	-	-	-	-
DO	•	•	•	•	•	•	•	•
DRFOF	•	•	•	•	•	•	•	•
DRIVE	•	•	•	•	•	•	•	•
DRIVEA	•	•	•	•	•	•	•	•
DYNFINISH	•	•	•	•	•	•	•	•
DYNNORM	•	•	•	•	•	•	•	•
DYNPOS	•	•	•	•	•	•	•	•
DYNROUGH	•	•	•	•	•	•	•	•
DYNSEMIFIN	•	•	•	•	•	•	•	•
DZERO	•	•	•	•	•	•	•	•
EAUTO	○	○	○	○	○	○	○	○
EGDEF	-	○	-	○	-	-	○	○
EGDEL	-	○	-	○	-	-	○	○
EGOFC	-	○	-	○	-	-	○	○
EGOFS	-	○	-	○	-	-	○	○
EGON	-	○	-	○	-	-	○	○
EGONSYN	-	○	-	○	-	-	○	○
EGONSYNE	-	○	-	○	-	-	○	○
ELSE	•	•	•	•	•	•	•	•
Enable	•	•	•	•	•	•	•	•
ENAT	○	○	○	○	○	○	○	○
ENDFOR	•	•	•	•	•	•	•	•
ENDIF	•	•	•	•	•	•	•	•
ENDLABEL	•	•	•	•	•	•	•	•
ENDLOOP	•	•	•	•	•	•	•	•
ENDPROC	•	•	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
ENDWHILE	•	•	•	•	•	•	•	•
ESRR	○	○	○	○	○	○	○	○
ESRS	○	○	○	○	○	○	○	○
ETAN	○	○	○	○	○	○	○	○
EVERY	•	•	•	•	•	•	•	•
EX	•	•	•	•	•	•	•	•
EXECSTRING	•	•	•	•	•	•	•	•
EXECTAB	•	•	•	•	•	•	•	•
EXECUTE	•	•	•	•	•	•	•	•
EXP	•	•	•	•	•	•	•	•
EXTCALL	•	•	•	•	•	•	•	•
EXTCLOSE	•	•	•	•	•	•	•	•
EXTERN	•	•	•	•	•	•	•	•
EXTOPEN	•	•	•	•	•	•	•	•
F	•	•	•	•	•	•	•	•
FA	•	•	•	•	•	•	•	•
FAD	•	•	•	•	•	•	•	•
FALSE	•	•	•	•	•	•	•	•
FB	•	•	•	•	•	•	•	•
FCTDEF	•	•	•	•	•	•	•	•
FCUB	•	•	•	•	•	•	•	•
FD	•	•	•	•	•	•	•	•
FDA	•	•	•	•	•	•	•	•
FENDNORM	•	•	•	•	•	•	•	•
FFWOF	•	•	•	•	•	•	•	•
FFWON	•	•	•	•	•	•	•	•
FGREF	•	•	•	•	•	•	•	•
FGROUP	•	•	•	•	•	•	•	•
FI	•	•	•	•	•	•	•	•
FIFOCTRL	•	•	•	•	•	•	•	•
FILEDATE	•	•	•	•	•	•	•	•
FILEINFO	•	•	•	•	•	•	•	•
FILESIZE	•	•	•	•	•	•	•	•
FILESTAT	•	•	•	•	•	•	•	•
FILETIME	•	•	•	•	•	•	•	•
FINEA	•	•	•	•	•	•	•	•
FL	•	•	•	•	•	•	•	•

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
FLIN	•	•	•	•	•	•	•	•
FMA	•	•	•	•	•	•	•	•
FNORM	•	•	•	•	•	•	•	•
FOCOF	○	○	○	○	○	○	○	○
FOCON	○	○	○	○	○	○	○	○
FOR	•	•	•	•	•	•	•	•
FP	•	•	•	•	•	•	•	•
FPO	-	-	-	-	-	-	-	-
FPR	•	•	•	•	•	•	•	•
FPRAOF	•	•	•	•	•	•	•	•
FPRAON	•	•	•	•	•	•	•	•
FRAME	•	•	•	•	•	•	•	•
FRC	•	•	•	•	•	•	•	•
FRCM	•	•	•	•	•	•	•	•
FROM	•	•	•	•	•	•	•	•
FTOC	•	•	•	•	•	•	•	•
FTOCOF	•	•	•	•	•	•	•	•
FTOCON	•	•	•	•	•	•	•	•
FXS	•	•	•	•	•	•	•	•
FXST	•	•	•	•	•	•	•	•
FXSW	•	•	•	•	•	•	•	•
FZ	•	•	•	•	•	•	•	•

Операторы G ... L

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
G0	•	•	•	•	•	•	•	•
G1	•	•	•	•	•	•	•	•
G2	•	•	•	•	•	•	•	•
G3	•	•	•	•	•	•	•	•
G4	•	•	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
G5	•	•	•	•	•	•	•	•
G7	•	•	•	•	•	•	•	•
G9	•	•	•	•	•	•	•	•
G17	•	•	•	•	•	•	•	•
G18	•	•	•	•	•	•	•	•
G19	•	•	•	•	•	•	•	•
G25	•	•	•	•	•	•	•	•
G26	•	•	•	•	•	•	•	•
G33	•	•	•	•	•	•	•	•
G34	•	•	•	•	•	•	•	•
G35	•	•	•	•	•	•	•	•
G40	•	•	•	•	•	•	•	•
G41	•	•	•	•	•	•	•	•
G42	•	•	•	•	•	•	•	•
G53	•	•	•	•	•	•	•	•
G54	•	•	•	•	•	•	•	•
G55	•	•	•	•	•	•	•	•
G56	•	•	•	•	•	•	•	•
G57	•	•	•	•	•	•	•	•
G58	→ G505							
G59	→ G506							
G60	•	•	•	•	•	•	•	•
G62	•	•	•	•	•	•	•	•
G63	•	•	•	•	•	•	•	•
G64	•	•	•	•	•	•	•	•
G70	•	•	•	•	•	•	•	•
G71	•	•	•	•	•	•	•	•
G74	•	•	•	•	•	•	•	•
G75	•	•	•	•	•	•	•	•
G90	•	•	•	•	•	•	•	•
G91	•	•	•	•	•	•	•	•
G93	•	•	•	•	•	•	•	•
G94	•	•	•	•	•	•	•	•
G95	•	•	•	•	•	•	•	•
G96	•	•	•	•	•	•	•	•
G97	•	•	•	•	•	•	•	•
G110	•	•	•	•	•	•	•	•

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
G111	•	•	•	•	•	•	•	•
G112	•	•	•	•	•	•	•	•
G140	•	•	•	•	•	•	•	•
G141	•	•	•	•	•	•	•	•
G142	•	•	•	•	•	•	•	•
G143	•	•	•	•	•	•	•	•
G147	•	•	•	•	•	•	•	•
G148	•	•	•	•	•	•	•	•
G153	•	•	•	•	•	•	•	•
G247	•	•	•	•	•	•	•	•
G248	•	•	•	•	•	•	•	•
G290	•	•	•	•	•	•	•	•
G291	•	•	•	•	•	•	•	•
G331	•	•	•	•	•	•	•	•
G332	•	•	•	•	•	•	•	•
G335	•	•	•	•	•	•	•	•
G336	•	•	•	•	•	•	•	•
G340	•	•	•	•	•	•	•	•
G341	•	•	•	•	•	•	•	•
G347	•	•	•	•	•	•	•	•
G348	•	•	•	•	•	•	•	•
G450	•	•	•	•	•	•	•	•
G451	•	•	•	•	•	•	•	•
G460	•	•	•	•	•	•	•	•
G461	•	•	•	•	•	•	•	•
G462	•	•	•	•	•	•	•	•
G500	•	•	•	•	•	•	•	•
G505 ... G599	•	•	•	•	•	•	•	•
G601	•	•	•	•	•	•	•	•
G602	•	•	•	•	•	•	•	•
G603	•	•	•	•	•	•	•	•
G621	•	•	•	•	•	•	•	•
G641	•	•	•	•	•	•	•	•
G642	•	•	•	•	•	•	•	•
G643	•	•	•	•	•	•	•	•
G644	•	•	•	•	•	•	•	•
G645	•	•	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
G700	•	•	•	•	•	•	•	•
G710	•	•	•	•	•	•	•	•
G810 ... G819	-	-	-	-	-	-	-	-
G820 ... G829	-	-	-	-	-	-	-	-
G931	•	•	•	•	•	•	•	•
G942	•	•	•	•	•	•	•	•
G952	•	•	•	•	•	•	•	•
G961	•	•	•	•	•	•	•	•
G962	•	•	•	•	•	•	•	•
G971	•	•	•	•	•	•	•	•
G972	•	•	•	•	•	•	•	•
G973	•	•	•	•	•	•	•	•
GEOAX	•	•	•	•	•	•	•	•
GET	•	•	•	•	•	•	•	•
GETACTT	•	•	•	•	•	•	•	•
GETACTTD	•	•	•	•	•	•	•	•
GETD	-	-	-	-	-	○	-	•
GETDNO	•	•	•	•	•	•	•	•
GETEXET	•	•	•	•	•	•	•	•
GETFREELOC	•	•	•	•	•	•	•	•
GETSELT	•	•	•	•	•	•	•	•
GETT	•	•	•	•	•	•	•	•
GETTCOR	•	•	•	•	•	•	•	•
GETTENV	•	•	•	•	•	•	•	•
GETVARAP	•	•	•	•	•	•	•	•
GETVARDFT	•	•	•	•	•	•	•	•
GETVARLIM	•	•	•	•	•	•	•	•
GETVARPHU	•	•	•	•	•	•	•	•
GETVARTYP	•	•	•	•	•	•	•	•
GFRAME0 ... GFRAME100	-	-	-	-	-	-	-	-
GOTO	•	•	•	•	•	•	•	•
GTOB	•	•	•	•	•	•	•	•
GTOC	•	•	•	•	•	•	•	•
GTOF	•	•	•	•	•	•	•	•
GOTOS	•	•	•	•	•	•	•	•
GP	•	•	•	•	•	•	•	•

Оператор	SINUMERIK 828D							
	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
GWPSOF	•	•	•	•	•	•	•	•
GROUP_	•	•	•	•	•	•	•	•
ADDEND								
GROUP_BEGIN	•	•	•	•	•	•	•	•
GROUP_END	•	•	•	•	•	•	•	•
GWPSON	•	•	•	•	•	•	•	•
H...	•	•	•	•	•	•	•	•
HOLES1	•	•	•	•	•	•	•	•
HOLES2	•	•	•	•	•	•	•	•
I	•	•	•	•	•	•	•	•
I1	•	•	•	•	•	•	•	•
IC	•	•	•	•	•	•	•	•
ICYCOF	•	•	•	•	•	•	•	•
ICYCON	•	•	•	•	•	•	•	•
ID	•	•	•	•	•	•	•	•
IDS	•	•	•	•	•	•	•	•
IF	•	•	•	•	•	•	•	•
INDEX	•	•	•	•	•	•	•	•
INIPO	•	•	•	•	•	•	•	•
INIRE	•	•	•	•	•	•	•	•
INICF	•	•	•	•	•	•	•	•
INIT	-	-	-	-	-	○	-	•
INITIAL								
INT	•	•	•	•	•	•	•	•
INTERSEC	•	•	•	•	•	•	•	•
INVCCW	-	-	-	-	-	-	-	-
INVCW	-	-	-	-	-	-	-	-
INVFRAME	•	•	•	•	•	•	•	•
IP	•	•	•	•	•	•	•	•
IPOBRKA	•	•	•	•	•	•	•	•
IPOENDA	•	•	•	•	•	•	•	•
IPTRLOCK	•	•	•	•	•	•	•	•
IPTRUNLOCK	•	•	•	•	•	•	•	•
IR	•	•	•	•	•	•	•	•
ISAXIS	•	•	•	•	•	•	•	•
ISD	-	-	-	-	-	-	-	-
ISFILE	•	•	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
ISNUMBER	•	•	•	•	•	•	•	•
ISOCALL	•	•	•	•	•	•	•	•
ISVAR	•	•	•	•	•	•	•	•
J	•	•	•	•	•	•	•	•
J1	•	•	•	•	•	•	•	•
JERKA	•	•	•	•	•	•	•	•
JERKLIM	•	•	•	•	•	•	•	•
JERKLIMA	•	•	•	•	•	•	•	•
JR	•	•	•	•	•	•	•	•
K	•	•	•	•	•	•	•	•
K1	•	•	•	•	•	•	•	•
KONT	•	•	•	•	•	•	•	•
KONTC	•	•	•	•	•	•	•	•
KONTT	•	•	•	•	•	•	•	•
KR	•	•	•	•	•	•	•	•
L	•	•	•	•	•	•	•	•
LEAD								
Ориентация ин- струмента	-	-	-	-	-	-	-	-
Полин. ориента- ции	-	-	-	-	-	-	-	-
LEADOF	-	-	-	•	-	-	•	•
LEADON	-	-	-	•	-	-	•	•
LENTOAX	•	•	•	•	•	•	•	•
LFOF	•	•	•	•	•	•	•	•
LFON	•	•	•	•	•	•	•	•
LFPOS	•	•	•	•	•	•	•	•
LFTXT	•	•	•	•	•	•	•	•
LFWP	•	•	•	•	•	•	•	•
LIFTFAST	•	•	•	•	•	•	•	•
LIMS	•	•	•	•	•	•	•	•
LLI	•	•	•	•	•	•	•	•
LN	•	•	•	•	•	•	•	•
LOCK	•	•	•	•	•	•	•	•
LONGHOLE	•	•	•	•	•	•	•	•
LOOP	•	•	•	•	•	•	•	•

Операторы М ... R

Оператор	SINUMERIK 828D							
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (m42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (m62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (m821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (m822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
M0	●	●	●	●	●	●	●	●
M1	●	●	●	●	●	●	●	●
M2	●	●	●	●	●	●	●	●
M3	●	●	●	●	●	●	●	●
M4	●	●	●	●	●	●	●	●
M5	●	●	●	●	●	●	●	●
M6	●	●	●	●	●	●	●	●
M17	●	●	●	●	●	●	●	●
M19	●	●	●	●	●	●	●	●
M30	●	●	●	●	●	●	●	●
M40	●	●	●	●	●	●	●	●
M41 ... M45	●	●	●	●	●	●	●	●
M70	●	●	●	●	●	●	●	●
MASLDEF	-	-	-	-	-	-	-	-
MASLDEL	-	-	-	-	-	-	-	-
MASLOF	-	-	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-	-	-
MASLON	-	-	-	-	-	-	-	-
MATCH	●	●	●	●	●	●	●	●
MAXVAL	●	●	●	●	●	●	●	●
MCALL	●	●	●	●	●	●	●	●
MEAC	-	-	-	-	-	-	-	-
MEAFRAME	●	●	●	●	●	●	●	●
MEAS	●	●	●	●	●	●	●	●
MEASA	-	-	-	-	-	-	-	-
MEASURE	●	●	●	●	●	●	●	●
MEAW	●	●	●	●	●	●	●	●
MEAWA	-	-	-	-	-	-	-	-
MI	●	●	●	●	●	●	●	●
MINDEX	●	●	●	●	●	●	●	●
MINVAL	●	●	●	●	●	●	●	●
MIRROR	●	●	●	●	●	●	●	●
MMC	●	●	●	●	●	●	●	●
MOD	●	●	●	●	●	●	●	●
MODAXVAL	●	●	●	●	●	●	●	●

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
MOV	•	•	•	•	•	•	•	•
MOVT	•	•	•	•	•	•	•	•
MSG	•	•	•	•	•	•	•	•
MVTOOL	•	•	•	•	•	•	•	•
N	•	•	•	•	•	•	•	•
NAMETOINT	•	•	•	•	•	•	•	•
ЧПУ	•	•	•	•	•	•	•	•
NEWCONF	•	•	•	•	•	•	•	•
NEWMT	•	•	•	•	•	•	•	•
NEWT	•	•	•	•	•	•	•	•
NORM	•	•	•	•	•	•	•	•
NOT	•	•	•	•	•	•	•	•
NPROT	•	•	•	•	•	•	•	•
NPROTDEF	•	•	•	•	•	•	•	•
NUMBER	•	•	•	•	•	•	•	•
OEMIPO1	-	-	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-	-	-
OF	•	•	•	•	•	•	•	•
OFFN	•	•	•	•	•	•	•	•
OMA1	-	-	-	-	-	-	-	-
OMA2	-	-	-	-	-	-	-	-
OMA3	-	-	-	-	-	-	-	-
OMA4	-	-	-	-	-	-	-	-
OMA5	-	-	-	-	-	-	-	-
OR	•	•	•	•	•	•	•	•
ORIXES	-	-	-	-	-	-	-	-
ORIXPOS	-	-	-	-	-	-	-	-
ORIC	-	-	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-	-	-
ORID	-	-	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-	-	-

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
ORIPATHS	-	-	-	-	-	-	-	-
ORIPANE	-	-	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-	-	-
ORIS	-	-	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-	-	-
ORISON	-	-	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-	-	-
OS	-	-	-	-	-	-	-	-
OSB	-	-	-	-	-	-	-	-
OSC	-	-	-	-	-	-	-	-
OSCILL	-	-	-	-	-	-	-	-
OSCTRL	-	-	-	-	-	-	-	-
OSD	-	-	-	-	-	-	-	-
OSE	-	-	-	-	-	-	-	-
OSNSC	•	•	•	•	•	•	•	•
OSOF	-	-	-	-	-	-	-	-
OSP1	•	•	•	•	•	•	•	•
OSP2	•	•	•	•	•	•	•	•
OSS	-	-	-	-	-	-	-	-
OSSE	-	-	-	-	-	-	-	-
OST	-	-	-	-	-	-	-	-
OST1	•	•	•	•	•	•	•	•
OST2	•	•	•	•	•	•	•	•
OTOL	•	•	•	•	•	•	•	•
OVR	•	•	•	•	•	•	•	•
OVRA	•	•	•	•	•	•	•	•
OVRRAP	•	•	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
P	•	•	•	•	•	•	•	•
PAROT	•	•	•	•	•	•	•	•
PAROTOF	•	•	•	•	•	•	•	•
PCALL	•	•	•	•	•	•	•	•
PDELAYOF	-	-	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-	-	-
PHI	-	-	-	-	-	-	-	-
PHU	•	•	•	•	•	•	•	•
PL	-	-	-	-	-	-	-	-
PM	•	•	•	•	•	•	•	•
po	-	-	-	-	-	-	-	-
POCKET3	•	•	•	•	•	•	•	•
POCKET4	•	•	•	•	•	•	•	•
POLF	•	•	•	•	•	•	•	•
POLFA	•	•	•	•	•	•	•	•
POLFMASK	•	•	•	•	•	•	•	•
POLFMLIN	•	•	•	•	•	•	•	•
POLY	-	-	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-	-	-
PON	-	-	-	-	-	-	-	-
PONS	-	-	-	-	-	-	-	-
Pos	•	•	•	•	•	•	•	•
POSA	•	•	•	•	•	•	•	•
POSM	•	•	•	•	•	•	•	•
POSMT	•	•	•	•	•	•	•	•
POSP	•	•	•	•	•	•	•	•
POSRANGE	•	•	•	•	•	•	•	•
POT	•	•	•	•	•	•	•	•
PR	•	•	•	•	•	•	•	•
PREPRO	•	•	•	•	•	•	•	•
PRESETON	•	•	•	•	•	•	•	•
PRESETONS	•	•	•	•	•	•	•	•
Prio	•	•	•	•	•	•	•	•
PRLOC	•	•	•	•	•	•	•	•
PROC	•	•	•	•	•	•	•	•
PROTA	•	•	•	•	•	•	•	•
PROTD	•	•	•	•	•	•	•	•

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
PROTS	•	•	•	•	•	•	•	•
PSI	-	-	-	-	-	-	-	-
PTP	•	•	•	•	•	•	•	•
PTPG0	•	•	•	•	•	•	•	•
PTPWOC	•	•	•	•	•	•	•	•
PUNCHACC	-	-	-	-	-	-	-	-
PUTFTOC	•	•	•	•	•	•	•	•
PUTFTOCF	•	•	•	•	•	•	•	•
PW	○	○	○	○	○	○	○	○
Qu	•	•	•	•	•	•	•	•
R ...	•	•	•	•	•	•	•	•
RAC	•	•	•	•	•	•	•	•
RDISABLE	•	•	•	•	•	•	•	•
READ	•	•	•	•	•	•	•	•
REAL	•	•	•	•	•	•	•	•
RELEASE	•	•	•	•	•	•	•	•
REP	•	•	•	•	•	•	•	•
REPEAT	•	•	•	•	•	•	•	•
REPEATB	•	•	•	•	•	•	•	•
REPOSA	•	•	•	•	•	•	•	•
REPOSH	•	•	•	•	•	•	•	•
REPOSHA	•	•	•	•	•	•	•	•
REPOSL	•	•	•	•	•	•	•	•
REPOSQ	•	•	•	•	•	•	•	•
REPOSQA	•	•	•	•	•	•	•	•
RESETMON	•	•	•	•	•	•	•	•
RET	•	•	•	•	•	•	•	•
RETB	•	•	•	•	•	•	•	•
RIC	•	•	•	•	•	•	•	•
RINDEX	•	•	•	•	•	•	•	•
RMB	•	•	•	•	•	•	•	•
RME	•	•	•	•	•	•	•	•
RMI	•	•	•	•	•	•	•	•
RMN	•	•	•	•	•	•	•	•
RND	•	•	•	•	•	•	•	•
RNDM	•	•	•	•	•	•	•	•
ROT	•	•	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
ROTS	•	•	•	•	•	•	•	•
ROUND	•	•	•	•	•	•	•	•
ROUNDUP	•	•	•	•	•	•	•	•
RP	•	•	•	•	•	•	•	•
RPL	•	•	•	•	•	•	•	•
RT	•	•	•	•	•	•	•	•
RTLIOF	•	•	•	•	•	•	•	•
RTLION	•	•	•	•	•	•	•	•

Операторы S ... Z

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
S	•	•	•	•	•	•	•	•
Save	•	•	•	•	•	•	•	•
SBLOF	•	•	•	•	•	•	•	•
SBLON	•	•	•	•	•	•	•	•
sc	•	•	•	•	•	•	•	•
SCALE	•	•	•	•	•	•	•	•
SCC	•	•	•	•	•	•	•	•
SCPARA	•	•	•	•	•	•	•	•
SD	○	○	○	○	○	○	○	○
set	•	•	•	•	•	•	•	•
SETAL	•	•	•	•	•	•	•	•
SETDNO	•	•	•	•	•	•	•	•
SETINT	•	•	•	•	•	•	•	•
SETM	-	-	-	-	-	○	-	•
SETMS	•	•	•	•	•	•	•	•
SETMS(n)	•	•	•	•	•	•	•	•
SETMTH	•	•	•	•	•	•	•	•
SETPIECE	•	•	•	•	•	•	•	•
SETTA	•	•	•	•	•	•	•	•

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
SETTCOR	•	•	•	•	•	•	•	•
SETTIA	•	•	•	•	•	•	•	•
SF	•	•	•	•	•	•	•	•
SIN	•	•	•	•	•	•	•	•
SIRELAY	-	-	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-	-	-
SLOT1	•	•	•	•	•	•	•	•
SLOT2	•	•	•	•	•	•	•	•
SOFT	•	•	•	•	•	•	•	•
SOFTA	•	•	•	•	•	•	•	•
SON	-	-	-	-	-	-	-	-
SONS	-	-	-	-	-	-	-	-
SPATH	•	•	•	•	•	•	•	•
SPCOF	•	•	•	•	•	•	•	•
SPCON	•	•	•	•	•	•	•	•
SPI	•	•	•	•	•	•	•	•
SPIF1	-	-	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○	○	○
SPN	-	-	-	-	-	-	-	-
SPOF	-	-	-	-	-	-	-	-
SPOS	•	•	•	•	•	•	•	•
SPOSA	•	•	•	•	•	•	•	•
SPP	-	-	-	-	-	-	-	-
SPRINT	•	•	•	•	•	•	•	•
SQRT	•	•	•	•	•	•	•	•
SR	•	•	•	•	•	•	•	•
SRA	•	•	•	•	•	•	•	•
ST	•	•	•	•	•	•	•	•
STA	•	•	•	•	•	•	•	•
START	-	-	-	-	-	○	-	•
STARTFIFO	•	•	•	•	•	•	•	•
STAT	•	•	•	•	•	•	•	•
STOLF	•	•	•	•	•	•	•	•
STOPFIFO	•	•	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
STOPRE	•	•	•	•	•	•	•	•
STOPREOF	•	•	•	•	•	•	•	•
STRING	•	•	•	•	•	•	•	•
STRINGFELD	•	•	•	•	•	•	•	•
STRINGIS	•	•	•	•	•	•	•	•
STRLEN	•	•	•	•	•	•	•	•
SUBSTR	•	•	•	•	•	•	•	•
SUPA	•	•	•	•	•	•	•	•
SVC	•	•	•	•	•	•	•	•
SYNFCT	•	•	•	•	•	•	•	•
SYNR	•	•	•	•	•	•	•	•
SYNRW	•	•	•	•	•	•	•	•
SYNW	•	•	•	•	•	•	•	•
T	•	•	•	•	•	•	•	•
TAN	•	•	•	•	•	•	•	•
TANG	-	-	-	-	-	-	-	-
TANGDEL	-	-	-	-	-	-	-	-
TANGOF	-	-	-	-	-	-	-	-
TANGON	-	-	-	-	-	-	-	-
TCA (828D: _TCA)	•	•	•	•	•	•	•	•
TCARR	•	-	•	-	•	•	-	-
TCI	•	•	•	•	•	•	•	•
TCOABS	•	-	•	-	•	•	-	-
TCOFR	•	-	•	-	•	•	-	-
TCOFRX	•	-	•	-	•	•	-	-
TCOFRY	•	-	•	-	•	•	-	-
TCOFRZ	•	-	•	-	•	•	-	-
THETA	-	-	-	-	-	-	-	-
TILT	-	-	-	-	-	-	-	-
TLIFT	-	-	-	-	-	-	-	-
TML	•	•	•	•	•	•	•	•
TMOF	•	•	•	•	•	•	•	•
TMON	•	•	•	•	•	•	•	•
TO	•	•	•	•	•	•	•	•
TOFF	•	•	•	•	•	•	•	•
TOFFL	•	•	•	•	•	•	•	•

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
TOFFOF	•	•	•	•	•	•	•	•
TOFFON	•	•	•	•	•	•	•	•
TOFFR	•	•	•	•	•	•	•	•
TOFRAME	•	•	•	•	•	•	•	•
TOFRAMEX	•	•	•	•	•	•	•	•
TOFRAMEY	•	•	•	•	•	•	•	•
TOFRAMEZ	•	•	•	•	•	•	•	•
TOLOWER	•	•	•	•	•	•	•	•
TOOLENV	•	•	•	•	•	•	•	•
TOOLGNT	•	•	•	•	•	•	•	•
TOOLGT	•	•	•	•	•	•	•	•
TOROT	•	•	•	•	•	•	•	•
TOROTOF	•	•	•	•	•	•	•	•
TOROTX	•	•	•	•	•	•	•	•
TOROTY	•	•	•	•	•	•	•	•
TOROTZ	•	•	•	•	•	•	•	•
TOUPPER	•	•	•	•	•	•	•	•
TOWBCS	•	-	•	-	•	•	-	-
TOWKCS	•	-	•	-	•	•	-	-
TOWMCS	•	-	•	-	•	•	-	-
TOWSTD	•	-	•	-	•	•	-	-
TOWTCS	•	-	•	-	•	•	-	-
TOWWCS	•	-	•	-	•	•	-	-
TR	•	•	•	•	•	•	•	•
TRAANG	-	-	-	-	-	-	○	○
TRACON	-	-	-	-	-	-	○	○
TRACYL	○	○	○	○	○	○	○	○
TRAFOOF	•	•	•	•	•	•	•	•
TRAFOON	-	-	-	-	-	-	-	-
TRAILOF	•	•	•	•	•	•	•	•
TRAILON	•	•	•	•	•	•	•	•
TRANS	•	•	•	•	•	•	•	•
Transmit	○	○	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-	-	-
TRUE	•	•	•	•	•	•	•	•
TRUNC	•	•	•	•	•	•	•	•
Tu	•	•	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D							
	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
TURN	•	•	•	•	•	•	•	•
ULI	•	•	•	•	•	•	•	•
UNLOCK	•	•	•	•	•	•	•	•
UNTIL	•	•	•	•	•	•	•	•
UPATH	•	•	•	•	•	•	•	•
Var	•	•	•	•	•	•	•	•
VELOLIM	•	•	•	•	•	•	•	•
VELOLIMA	•	•	•	•	•	•	•	•
WAITC	•	•	•	•	•	•	•	•
WAITE	-	-	-	-	-	○	-	•
WAITENC	•	•	•	•	•	•	•	•
WAITM	-	-	-	-	-	○	-	•
WAITMC	-	-	-	-	-	○	-	•
WAITP	•	•	•	•	•	•	•	•
WAITS	•	•	•	•	•	•	•	•
WALCS0	•	•	•	•	•	•	•	•
WALCS1	•	•	•	•	•	•	•	•
WALCS2	•	•	•	•	•	•	•	•
WALCS3	•	•	•	•	•	•	•	•
WALCS4	•	•	•	•	•	•	•	•
WALCS5	•	•	•	•	•	•	•	•
WALCS6	•	•	•	•	•	•	•	•
WALCS7	•	•	•	•	•	•	•	•
WALCS8	•	•	•	•	•	•	•	•
WALCS9	•	•	•	•	•	•	•	•
WALCS10	•	•	•	•	•	•	•	•
WALIMOF	•	•	•	•	•	•	•	•
WALIMON	•	•	•	•	•	•	•	•
WHEN	•	•	•	•	•	•	•	•
WHENEVER	•	•	•	•	•	•	•	•
WHILE	•	•	•	•	•	•	•	•
WORKPIECE	•	•	•	•	•	•	•	•
write	•	•	•	•	•	•	•	•
WRTPR	•	•	•	•	•	•	•	•
X	•	•	•	•	•	•	•	•
XOR	•	•	•	•	•	•	•	•

Оператор	SINUMERIK 828D							
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW Фрезеро- вание Экспорт (me42)	SW24x(5) CNC-SW Токарная обработ- ка Экспорт (te42)	SW26x(3) CNC-SW Фрезеро- вание Экспорт (me62)	SW26x(3) CNC-SW Токарная обработ- ка Экспорт (te62)	SW28x(2) CNC-SW Фрезеро- вание Экспорт (me821)	SW28x(1) CNC-SW Фрезерова- ние Расшир. Экспорт (me822)	SW28x(2) CNC-SW Токарная обработ- ка Экспорт (te821)	SW28x(1) CNC-SW Токарная обработка Расшир. Экспорт (te822)
Y	•	•	•	•	•	•	•	•
Z	•	•	•	•	•	•	•	•

21.2.2 Варианты управления: шлифование

Операторы A ... C

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
:	●	●	●	●	●	●
*	●	●	●	●	●	●
+	●	●	●	●	●	●
-	●	●	●	●	●	●
<	●	●	●	●	●	●
<<	●	●	●	●	●	●
<=	●	●	●	●	●	●
=	●	●	●	●	●	●
>=	●	●	●	●	●	●
/	●	●	●	●	●	●
/0 ... /7	●	●	●	●	●	●
A	●	●	●	●	●	●
A2	-	-	-	-	-	-
A3	-	-	-	-	-	-
A4	-	-	-	-	-	-
A5	-	-	-	-	-	-
A6	-	-	-	-	-	-
A7	-	-	-	-	-	-
ABS	●	●	●	●	●	●
AC	●	●	●	●	●	●
ACC	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
ACN	●	●	●	●	●	●
ACOS	●	●	●	●	●	●
ACP	●	●	●	●	●	●
ACTBLOCNO	●	●	●	●	●	●
ADDFRAME	●	●	●	●	●	●
ADIS	●	●	●	●	●	●
ADISPOS	●	●	●	●	●	●
ADISPOSA	●	●	●	●	●	●
ALF	●	●	●	●	●	●
AMIRROR	●	●	●	●	●	●
AND	●	●	●	●	●	●
ANG	●	●	●	●	●	●
AP	●	●	●	●	●	●
APR	●	●	●	●	●	●
APRB	●	●	●	●	●	●
APRP	●	●	●	●	●	●
APW	●	●	●	●	●	●
APWB	●	●	●	●	●	●
APWP	●	●	●	●	●	●
APX	●	●	●	●	●	●
AR	●	●	●	●	●	●
AROT	●	●	●	●	●	●
AROTS	●	●	●	●	●	●
AS	●	●	●	●	●	●
ASCALE	●	●	●	●	●	●
ASIN	●	●	●	●	●	●
ASPLINE	○	○	○	○	○	○
ATAN2	●	●	●	●	●	●
ATOL	●	●	●	●	●	●
ATRANS	●	●	●	●	●	●
AUXFUDEL	●	●	●	●	●	●
AUXFUDELG	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●
AX	●	●	●	●	●	●
AXCTSWE	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
● стандарт ○ опция - недоступно						
AXIS	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●
B	●	●	●	●	●	●
B2	-	-	-	-	-	-
B3	-	-	-	-	-	-
B4	-	-	-	-	-	-
B5	-	-	-	-	-	-
B6	-	-	-	-	-	-
B7	-	-	-	-	-	-
B_AND	●	●	●	●	●	●
B_OR	●	●	●	●	●	●
B_NOT	●	●	●	●	●	●
B_XOR	●	●	●	●	●	●
BAUTO	○	○	○	○	○	○
BLOCK	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●
BNAT	○	○	○	○	○	○
BOOL	●	●	●	●	●	●
BOUND	●	●	●	●	●	●
BRISK	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●
BSPLINE	○	○	○	○	○	○
BTAN	○	○	○	○	○	○
C	●	●	●	●	●	●
C2	-	-	Имя оси кана- ла	-	-	-
C3	-	-	-	-	-	-
C4	-	-	-	-	-	-
C5	-	-	-	-	-	-
C6	-	-	-	-	-	-
C7	-	-	-	-	-	-
CAC	●	●	●	●	●	●
CACN	●	●	●	●	●	●
CACP	●	●	●	●	●	●
CALCDAT	●	●	●	●	●	●

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
• стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
CALCPOSI	•	•	•	•	•	•
CALL	•	•	•	•	•	•
CALLPATH	•	•	•	•	•	•
CANCEL	•	•	•	•	•	•
CASE	•	•	•	•	•	•
CDC	•	•	•	•	•	•
CDOF	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-
CDON	-	-	-	-	-	-
CFC	•	•	•	•	•	•
CFIN	•	•	•	•	•	•
CFINE	•	•	•	•	•	•
CFTCP	•	•	•	•	•	•
Chan	•	•	•	•	•	•
CHANDATA	•	•	•	•	•	•
CHAR	•	•	•	•	•	•
CHF	•	•	•	•	•	•
CHKDM	•	•	•	•	•	•
CHKDNO	•	•	•	•	•	•
CHR	•	•	•	•	•	•
CIC	•	•	•	•	•	•
CIP	•	•	•	•	•	•
CLEARM	-	-	•	-	-	•
CLRINT	•	•	•	•	•	•
CMIRROR	•	•	•	•	•	•
COARSEA	•	•	•	•	•	•
COLLPAIR	-	-	-	-	-	-
COMPCAD	•	•	•	•	•	•
COMPCURV	•	•	•	•	•	•
COMPLETE	•	•	•	•	•	•
COMPOF	•	•	•	•	•	•
COMPON	•	•	•	•	•	•
COMPSURF	-	-	-	-	-	-
CONTDCON	•	•	•	•	•	•
CONTPRON	•	•	•	•	•	•
CORROF	•	•	•	•	•	•
CORRTRAFO	-	-	-	-	-	-
COS	•	•	•	•	•	•

Оператор	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
● стандарт ○ опция - недоступно						
COUPDEF	○	○	○	○	○	○
COUPDEL	○	○	○	○	○	○
COUPOF	○	○	○	○	○	○
COUPOFS	○	○	○	○	○	○
COUPON	○	○	○	○	○	○
COUPONC	○	○	○	○	○	○
COUPRES	○	○	○	○	○	○
CP	●	●	●	●	●	●
CPBC	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○
CPFMOF	○	○	○	○	○	○
CPFMON	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○
CPFRS	○	○	○	○	○	○
CPLA	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○
CPLINTR	○	○	○	○	○	○
CPLNUM	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○
CPLON	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○
CPLOUTTR	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○
CPOF	○	○	○	○	○	○
CPON	○	○	○	○	○	○

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
CPRECOF	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●
CPRES	○	○	○	○	○	○
CPROT	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○
CPSYNCOF	○	○	○	○	○	○
CPSYNCOF2	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○
CR	●	●	●	●	●	●
CROT	●	●	●	●	●	●
CROTS	●	●	●	●	●	●
CRPL	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○
CT	●	●	●	●	●	●
CTAB	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-
CTABID	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-
CTABNO	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор ● стандарт ○ опция - недоступно	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
CTABPOLID	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-
CTABSEGID	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-
CTOL	●	●	●	●	●	●
CTRANS	●	●	●	●	●	●
CUT2D	●	●	●	●	●	●
CUT2DD	●	●	●	●	●	●
CUT2DF	●	●	●	●	●	●
CUT2DFD	●	●	●	●	●	●
CUT3DC	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●
CUTCONON	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●
CUTMODK	-	-	-	-	-	-
CYCLE60	-	-	-	-	-	-
CYCLE61	-	-	-	-	-	-
CYCLE62	●	●	●	●	●	●
CYCLE63	-	-	-	-	-	-
CYCLE64	-	-	-	-	-	-
CYCLE70	-	-	-	-	-	-
CYCLE72	-	-	-	-	-	-
CYCLE76	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
CYCLE77	-	-	-	-	-	-
CYCLE78	-	-	-	-	-	-
CYCLE79	-	-	-	-	-	-
CYCLE81	-	-	-	-	-	-
CYCLE82	-	-	-	-	-	-
CYCLE83	-	-	-	-	-	-
CYCLE84	-	-	-	-	-	-
CYCLE85	-	-	-	-	-	-
CYCLE86	-	-	-	-	-	-
CYCLE92	-	-	-	-	-	-
CYCLE95	-	-	-	-	-	-
CYCLE98	-	-	-	-	-	-
CYCLE99	-	-	-	-	-	-
CYCLE150	-	-	-	-	-	-
CYCLE435	○	○	○	○	○	○
CYCLE495	○	○	○	○	○	○
CYCLE750	●	●	●	●	●	●
CYCLE751	●	●	●	●	●	●
CYCLE752	●	●	●	●	●	●
CYCLE753	●	●	●	●	●	●
CYCLE754	●	●	●	●	●	●
CYCLE755	●	●	●	●	●	●
CYCLE756	●	●	●	●	●	●
CYCLE757	●	●	●	●	●	●
CYCLE758	●	●	●	●	●	●
CYCLE759	●	●	●	●	●	●
CYCLE800	○	○	○	○	○	○
CYCLE801	-	-	-	-	-	-
CYCLE802	-	-	-	-	-	-
CYCLE830	-	-	-	-	-	-
CYCLE832	●	●	●	●	●	●
CYCLE840	-	-	-	-	-	-
CYCLE899	-	-	-	-	-	-
CYCLE930	-	-	-	-	-	-
CYCLE940	-	-	-	-	-	-
CYCLE951	-	-	-	-	-	-
CYCLE952	-	-	-	-	-	-
CYCLE961	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
CYCLE971	-	-	-	-	-	-
CYCLE973	-	-	-	-	-	-
CYCLE974	-	-	-	-	-	-
CYCLE976	-	-	-	-	-	-
CYCLE977	-	-	-	-	-	-
CYCLE978	-	-	-	-	-	-
CYCLE979	-	-	-	-	-	-
CYCLE982	-	-	-	-	-	-
CYCLE994	-	-	-	-	-	-
CYCLE995	-	-	-	-	-	-
CYCLE996	-	-	-	-	-	-
CYCLE997	-	-	-	-	-	-
CYCLE998	-	-	-	-	-	-
CYCLE4071	•	•	•	•	•	•
CYCLE4072	•	•	•	•	•	•
CYCLE4073	•	•	•	•	•	•
CYCLE4074	•	•	•	•	•	•
CYCLE4075	•	•	•	•	•	•
CYCLE4077	•	•	•	•	•	•
CYCLE4078	•	•	•	•	•	•
CYCLE4079	•	•	•	•	•	•

Операторы D ... F

Оператор	SINUMERIK 828D					
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
D	•	•	•	•	•	•
D0	•	•	•	•	•	•
DAC	•	•	•	•	•	•
DC	•	•	•	•	•	•
DCI	•	•	•	•	•	•
DCM	•	•	•	•	•	•
DCU	•	•	•	•	•	•
DEF	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
DEFINE	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●
DELDL	●	●	●	●	●	●
DELDTG	●	●	●	●	●	●
DELETE	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●
DELMRES	●	●	●	●	●	●
DELMT	-	-	-	-	-	-
DELOBJ	-	-	-	-	-	-
DELT	●	●	●	●	●	●
DELTC	●	●	●	●	●	●
DELTOOLNV	●	●	●	●	●	●
DIACYCOFA	●	●	●	●	●	●
DIAM90	●	●	●	●	●	●
DIAM90A	●	●	●	●	●	●
DIAMCHAN	●	●	●	●	●	●
DIAMCHANA	●	●	●	●	●	●
DIAMCYCOF	●	●	●	●	●	●
DIAMOF	●	●	●	●	●	●
DIAMOFA	●	●	●	●	●	●
DIAMON	●	●	●	●	●	●
DIAMONA	●	●	●	●	●	●
DIC	●	●	●	●	●	●
DILF	●	●	●	●	●	●
Disable	●	●	●	●	●	●
DISC	●	●	●	●	●	●
DISCL	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●
DISPLON	●	●	●	●	●	●
DISPR	●	●	●	●	●	●
DISR	●	●	●	●	●	●
DISRP	●	●	●	●	●	●
DITE	●	●	●	●	●	●
DITS	●	●	●	●	●	●
DIV	●	●	●	●	●	●
dl	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор ● стандарт ○ опция - недоступно	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
DO	●	●	●	●	●	●
DRFOF	●	●	●	●	●	●
DRIVE	●	●	●	●	●	●
DRIVEA	●	●	●	●	●	●
DYNFINISH	●	●	●	●	●	●
DYNNORM	●	●	●	●	●	●
DYNPOS	●	●	●	●	●	●
DYNROUGH	●	●	●	●	●	●
DYNSEMIFIN	●	●	●	●	●	●
DZERO	●	●	●	●	●	●
EAUTO	○	○	○	○	○	○
EGDEF	○	○	○	○	○	○
EGDEL	○	○	○	○	○	○
EGOFC	○	○	○	○	○	○
EGOFS	○	○	○	○	○	○
EGON	○	○	○	○	○	○
EGONSYN	○	○	○	○	○	○
EGONSYNE	○	○	○	○	○	○
ELSE	●	●	●	●	●	●
Enable	●	●	●	●	●	●
ENAT	○	○	○	○	○	○
ENDFOR	●	●	●	●	●	●
ENDIF	●	●	●	●	●	●
ENDLABEL	●	●	●	●	●	●
ENDLOOP	●	●	●	●	●	●
ENDPROC	●	●	●	●	●	●
ENDWHILE	●	●	●	●	●	●
ESRR	○	○	○	○	○	○
ESRS	○	○	○	○	○	○
ETAN	○	○	○	○	○	○
EVERY	●	●	●	●	●	●
EX	●	●	●	●	●	●
EXECSTRING	●	●	●	●	●	●
EXECTAB	●	●	●	●	●	●
EXECUTE	●	●	●	●	●	●
exp	●	●	●	●	●	●
EXTCALL	●	●	●	●	●	●
EXTCLOSE	●	●	●	●	●	●

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
EXTERN	●	●	●	●	●	●
EXTOPEN	●	●	●	●	●	●
F	●	●	●	●	●	●
FA	●	●	●	●	●	●
FAD	●	●	●	●	●	●
false	●	●	●	●	●	●
FB	●	●	●	●	●	●
FCTDEF	●	●	●	●	●	●
FCUB	●	●	●	●	●	●
FD	●	●	●	●	●	●
FDA	●	●	●	●	●	●
FENDNORM	●	●	●	●	●	●
FFWOF	●	●	●	●	●	●
FFWON	●	●	●	●	●	●
FGREF	●	●	●	●	●	●
FGROUP	●	●	●	●	●	●
FI	●	●	●	●	●	●
FIFOCTRL	●	●	●	●	●	●
FILEDATE	●	●	●	●	●	●
FILEINFO	●	●	●	●	●	●
FILESIZE	●	●	●	●	●	●
FILESTAT	●	●	●	●	●	●
FILETIME	●	●	●	●	●	●
FINEA	●	●	●	●	●	●
FL	●	●	●	●	●	●
FLIN	●	●	●	●	●	●
FMA	●	●	●	●	●	●
FNORM	●	●	●	●	●	●
FOCOF	○	○	○	○	○	○
FOCON	○	○	○	○	○	○
FOR	●	●	●	●	●	●
FP	●	●	●	●	●	●
FPO	-	-	-	-	-	-
FPR	●	●	●	●	●	●
FPRAOF	●	●	●	●	●	●
FPRAON	●	●	●	●	●	●
FRAME	●	●	●	●	●	●
FRC	●	●	●	●	●	●

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
FRCM	•	•	•	•	•	•
FROM	•	•	•	•	•	•
FTOC	•	•	•	•	•	•
FTOCOF	•	•	•	•	•	•
FTOCON	•	•	•	•	•	•
FXS	•	•	•	•	•	•
FXST	•	•	•	•	•	•
FXSW	•	•	•	•	•	•
FZ	•	•	•	•	•	•

Операторы G ... L

Оператор	SINUMERIK 828D					
• стандарт ○ опция - недоступно	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
G0	•	•	•	•	•	•
G1	•	•	•	•	•	•
G2	•	•	•	•	•	•
G3	•	•	•	•	•	•
G4	•	•	•	•	•	•
G5	•	•	•	•	•	•
G7	•	•	•	•	•	•
G9	•	•	•	•	•	•
G17	•	•	•	•	•	•
G18	•	•	•	•	•	•
G19	•	•	•	•	•	•
G25	•	•	•	•	•	•
G26	•	•	•	•	•	•
G33	•	•	•	•	•	•
G34	•	•	•	•	•	•
G35	•	•	•	•	•	•
G40	•	•	•	•	•	•
G41	•	•	•	•	•	•
G42	•	•	•	•	•	•
G53	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
G54	●	●	●	●	●	●
G55	●	●	●	●	●	●
G56	●	●	●	●	●	●
G57	●	●	●	●	●	●
G58	→ G505					
G59	→ G506					
G60	●	●	●	●	●	●
G62	●	●	●	●	●	●
G63	●	●	●	●	●	●
G64	●	●	●	●	●	●
G70	●	●	●	●	●	●
G71	●	●	●	●	●	●
G74	●	●	●	●	●	●
G75	●	●	●	●	●	●
G90	●	●	●	●	●	●
G91	●	●	●	●	●	●
G93	●	●	●	●	●	●
G94	●	●	●	●	●	●
G95	●	●	●	●	●	●
G96	●	●	●	●	●	●
G97	●	●	●	●	●	●
G110	●	●	●	●	●	●
G111	●	●	●	●	●	●
G112	●	●	●	●	●	●
G140	●	●	●	●	●	●
G141	●	●	●	●	●	●
G142	●	●	●	●	●	●
G143	●	●	●	●	●	●
G147	●	●	●	●	●	●
G148	●	●	●	●	●	●
G153	●	●	●	●	●	●
G247	●	●	●	●	●	●
G248	●	●	●	●	●	●
G290	●	●	●	●	●	●
G291	-	-	-	-	-	-
G331	●	●	●	●	●	●
G332	●	●	●	●	●	●
G335	●	●	●	●	●	●

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор ● стандарт ○ опция - недоступно	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
G336	●	●	●	●	●	●
G340	●	●	●	●	●	●
G341	●	●	●	●	●	●
G347	●	●	●	●	●	●
G348	●	●	●	●	●	●
G450	●	●	●	●	●	●
G451	●	●	●	●	●	●
G460	●	●	●	●	●	●
G461	●	●	●	●	●	●
G462	●	●	●	●	●	●
G500	●	●	●	●	●	●
G505 ... G599	●	●	●	●	●	●
G601	●	●	●	●	●	●
G602	●	●	●	●	●	●
G603	●	●	●	●	●	●
G621	●	●	●	●	●	●
G641	●	●	●	●	●	●
G642	●	●	●	●	●	●
G643	●	●	●	●	●	●
G644	●	●	●	●	●	●
G645	●	●	●	●	●	●
G700	●	●	●	●	●	●
G710	●	●	●	●	●	●
G810 ... G819	-	-	-	-	-	-
G820 ... G829	-	-	-	-	-	-
G931	●	●	●	●	●	●
G942	●	●	●	●	●	●
G952	●	●	●	●	●	●
G961	●	●	●	●	●	●
G962	●	●	●	●	●	●
G971	●	●	●	●	●	●
G972	●	●	●	●	●	●
G973	●	●	●	●	●	●
GEOAX	●	●	●	●	●	●
GET	●	●	●	●	●	●
GETACTT	●	●	●	●	●	●
GETACTTD	●	●	●	●	●	●
GETD	-	-	●	-	-	●

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
• стандарт						
○ опция						
- недоступно						
GETDNO	•	•	•	•	•	•
GETEXET	•	•	•	•	•	•
GETFREELOC	•	•	•	•	•	•
GETSELT	•	•	•	•	•	•
GETT	•	•	•	•	•	•
GETTCOR	•	•	•	•	•	•
GETTENV	•	•	•	•	•	•
GETVARAP	•	•	•	•	•	•
GETVARDFT	•	•	•	•	•	•
GETVARLIM	•	•	•	•	•	•
GETVARPHU	•	•	•	•	•	•
GETVARTYP	•	•	•	•	•	•
GFRAME0 ... GFRAME100	<50	< 100	< 100	<50	< 100	< 100
GOTO	•	•	•	•	•	•
GOTOB	•	•	•	•	•	•
GOTOC	•	•	•	•	•	•
GOTOF	•	•	•	•	•	•
GOTOS	•	•	•	•	•	•
GP	•	•	•	•	•	•
GWPSOF	•	•	•	•	•	•
GROUP_ADDEND	•	•	•	•	•	•
GROUP_BEGIN	•	•	•	•	•	•
GROUP_END	•	•	•	•	•	•
GWPSON	•	•	•	•	•	•
H...	•	•	•	•	•	•
HOLES1	-	-	-	-	-	-
HOLES2	-	-	-	-	-	-
I	•	•	•	•	•	•
I1	•	•	•	•	•	•
IC	•	•	•	•	•	•
ICYCOF	•	•	•	•	•	•
ICYCON	•	•	•	•	•	•
ID	•	•	•	•	•	•
IDS	•	•	•	•	•	•
IF	•	•	•	•	•	•
INDEX	•	•	•	•	•	•
INIPO	•	•	•	•	•	•

Оператор	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
• стандарт						
○ опция						
- недоступно						
INIRE	•	•	•	•	•	•
INICF	•	•	•	•	•	•
INIT	-	-	•	-	-	•
INITIAL						
INT	•	•	•	•	•	•
INTERSEC	•	•	•	•	•	•
INVCCW	-	-	-	-	-	-
INVCW	-	-	-	-	-	-
INVFRAME	•	•	•	•	•	•
IP	•	•	•	•	•	•
IPOBRKA	•	•	•	•	•	•
IPOENDA	•	•	•	•	•	•
IPTRLOCK	•	•	•	•	•	•
IPTRUNLOCK	•	•	•	•	•	•
IR	•	•	•	•	•	•
ISAXIS	•	•	•	•	•	•
ISD	-	-	-	-	-	-
ISFILE	•	•	•	•	•	•
ISNUMBER	•	•	•	•	•	•
ISOCALL	-	-	-	-	-	-
ISVAR	•	•	•	•	•	•
J	•	•	•	•	•	•
J1	•	•	•	•	•	•
JERKA	•	•	•	•	•	•
JERKLIM	•	•	•	•	•	•
JERKLIMA	•	•	•	•	•	•
JR	•	•	•	•	•	•
K	•	•	•	•	•	•
K1	•	•	•	•	•	•
KONT	•	•	•	•	•	•
KONTC	•	•	•	•	•	•
KONTT	•	•	•	•	•	•
KR	•	•	•	•	•	•
I	•	•	•	•	•	•
LEAD						
Ориентация инстру- мента	-	-	-	-	-	-
Полин. ориентации	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
LEADOF	-	-	-	-	-	-
LEADON	-	-	-	-	-	-
LENTOAX	●	●	●	●	●	●
LFOF	●	●	●	●	●	●
LFON	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●
LFWP	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●
LIMS	●	●	●	●	●	●
LLI	●	●	●	●	●	●
LN	●	●	●	●	●	●
LOCK	●	●	●	●	●	●
LONGHOLE	-	-	-	-	-	-
LOOP	●	●	●	●	●	●

Операторы M ... R

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
M0	●	●	●	●	●	●
M1	●	●	●	●	●	●
M2	●	●	●	●	●	●
M3	●	●	●	●	●	●
M4	●	●	●	●	●	●
M5	●	●	●	●	●	●
M6	●	●	●	●	●	●
M17	●	●	●	●	●	●
M19	●	●	●	●	●	●
M30	●	●	●	●	●	●
M40	●	●	●	●	●	●
M41 ... M45	●	●	●	●	●	●
M70	●	●	●	●	●	●
MASLDEF	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
MASLDEL	-	-	-	-	-	-
MASLOF	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-
MASLON	-	-	-	-	-	-
MATCH	•	•	•	•	•	•
MAXVAL	•	•	•	•	•	•
MCALL	•	•	•	•	•	•
MEAC	-	○	○	-	○	○
MEAFRAME	•	•	•	•	•	•
MEAS	•	•	•	•	•	•
MEASA	-	○	○	-	○	○
MEASURE	•	•	•	•	•	•
MEAW	•	•	•	•	•	•
MEAWA	-	○	○	-	○	○
MI	•	•	•	•	•	•
MINDEX	•	•	•	•	•	•
MINVAL	•	•	•	•	•	•
MIRROR	•	•	•	•	•	•
MMC	•	•	•	•	•	•
MOD	•	•	•	•	•	•
MODAXVAL	•	•	•	•	•	•
MOV	•	•	•	•	•	•
MOVT	•	•	•	•	•	•
MSG	•	•	•	•	•	•
MVTOOL	•	•	•	•	•	•
N	•	•	•	•	•	•
NAMETOINT	•	•	•	•	•	•
NCK	•	•	•	•	•	•
NEWCONF	•	•	•	•	•	•
NEWMT	•	•	•	•	•	•
NEWT	-	-	-	-	-	-
NORM	•	•	•	•	•	•
NOT	•	•	•	•	•	•
NPROT	•	•	•	•	•	•
NPROTDEF	•	•	•	•	•	•
NUMBER	•	•	•	•	•	•
OEMIPO1	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
OF	●	●	●	●	●	●
OFFN	●	●	●	●	●	●
OMA1	-	-	-	-	-	-
OMA2	-	-	-	-	-	-
OMA3	-	-	-	-	-	-
OMA4	-	-	-	-	-	-
OMA5	-	-	-	-	-	-
OR	●	●	●	●	●	●
ORIXES	-	-	-	-	-	-
ORIXPOS	-	-	-	-	-	-
ORIC	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-
ORID	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-
ORIPANE	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-
ORIS	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-
ORISON	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-
OS	○	○	○	○	○	○

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор ● стандарт ○ опция - недоступно	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
OSB	○	○	○	○	○	○
OSC	-	-	-	-	-	-
OSCILL	○	○	○	○	○	○
OSCTRL	○	○	○	○	○	○
OSD	-	-	-	-	-	-
OSE	○	○	○	○	○	○
OSNSC	●	●	●	●	●	●
OSOF	-	-	-	-	-	-
OSP1	●	●	●	●	●	●
OSP2	●	●	●	●	●	●
OSS	-	-	-	-	-	-
OSSE	-	-	-	-	-	-
OST	-	-	-	-	-	-
OST1	●	●	●	●	●	●
OST2	●	●	●	●	●	●
OTOL	●	●	●	●	●	●
OVR	●	●	●	●	●	●
OVRA	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●
P	●	●	●	●	●	●
PAROT	●	●	●	●	●	●
PAROTOF	●	●	●	●	●	●
PCALL	●	●	●	●	●	●
PDELAYOF	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-
PHI	-	-	-	-	-	-
PHU	●	●	●	●	●	●
PL	-	-	-	-	-	-
PM	●	●	●	●	●	●
po	-	-	-	-	-	-
POCKET3	-	-	-	-	-	-
POCKET4	-	-	-	-	-	-
POLF	●	●	●	●	●	●
POLFA	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●
POLY	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
PON	-	-	-	-	-	-
PONS	-	-	-	-	-	-
Pos	●	●	●	●	●	●
POSA	●	●	●	●	●	●
POSM	●	●	●	●	●	●
POSMT	-	-	-	-	-	-
POSP	●	●	●	●	●	●
POSRANGE	●	●	●	●	●	●
POT	●	●	●	●	●	●
PR	●	●	●	●	●	●
PREPRO	●	●	●	●	●	●
PRESETON	●	●	●	●	●	●
PRESETONS	●	●	●	●	●	●
Prio	●	●	●	●	●	●
PRLOC	●	●	●	●	●	●
PROC	●	●	●	●	●	●
PROTA	●	●	●	●	●	●
PROTD	●	●	●	●	●	●
PROTS	●	●	●	●	●	●
PSI	-	-	-	-	-	-
PTP	●	●	●	●	●	●
PTPG0	●	●	●	●	●	●
PTPWOC	●	●	●	●	●	●
PUNCHACC	-	-	-	-	-	-
PUTFTOC	●	●	●	●	●	●
PUTFTOCF	●	●	●	●	●	●
PW	○	○	○	○	○	○
Qu	●	●	●	●	●	●
R ...	●	●	●	●	●	●
RAC	●	●	●	●	●	●
RDISABLE	●	●	●	●	●	●
READ	●	●	●	●	●	●
REAL	●	●	●	●	●	●
RELEASE	●	●	●	●	●	●
REP	●	●	●	●	●	●
REPEAT	●	●	●	●	●	●
REPEATB	●	●	●	●	●	●
REPOSA	●	●	●	●	●	●

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
REPOSH	●	●	●	●	●	●
REPOSHA	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●
RET	●	●	●	●	●	●
RETB	●	●	●	●	●	●
RIC	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●
RMB	●	●	●	●	●	●
RME	●	●	●	●	●	●
RMI	●	●	●	●	●	●
RMN	●	●	●	●	●	●
RND	●	●	●	●	●	●
RNDM	●	●	●	●	●	●
ROT	●	●	●	●	●	●
ROTS	●	●	●	●	●	●
ROUND	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●
RP	●	●	●	●	●	●
RPL	●	●	●	●	●	●
RT	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●
RTLION	●	●	●	●	●	●

Операторы S ... Z

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
S	●	●	●	●	●	●
Save	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●
SBLON	●	●	●	●	●	●

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
• стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
sc	•	•	•	•	•	•
SCALE	•	•	•	•	•	•
SCC	•	•	•	•	•	•
SCPARA	•	•	•	•	•	•
SD	○	○	○	○	○	○
set	•	•	•	•	•	•
SETAL	•	•	•	•	•	•
SETDNO	•	•	•	•	•	•
SETINT	•	•	•	•	•	•
SETM	-	-	•	-	-	•
SETMS	•	•	•	•	•	•
SETMS(n)	•	•	•	•	•	•
SETMTH	•	•	•	•	•	•
SETPIECE	•	•	•	•	•	•
SETTA	•	•	•	•	•	•
SETTCOR	•	•	•	•	•	•
SETTIA	•	•	•	•	•	•
SF	•	•	•	•	•	•
SIN	•	•	•	•	•	•
SIRELAY	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-
SLOT1	-	-	-	-	-	-
SLOT2	-	-	-	-	-	-
SOFT	•	•	•	•	•	•
SOFTA	•	•	•	•	•	•
SON	-	-	-	-	-	-
SONS	-	-	-	-	-	-
SPATH	•	•	•	•	•	•
SPCOF	•	•	•	•	•	•
SPCON	•	•	•	•	•	•
SPI	•	•	•	•	•	•
SPIF1	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○
SPN	-	-	-	-	-	-
SPOF	-	-	-	-	-	-

Оператор	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
SPOS	•	•	•	•	•	•
SPOSA	•	•	•	•	•	•
SPP	-	-	-	-	-	-
SPRINT	•	•	•	•	•	•
SQRT	•	•	•	•	•	•
SR	•	•	•	•	•	•
SRA	•	•	•	•	•	•
ST	•	•	•	•	•	•
STA	•	•	•	•	•	•
START	-	-	•	-	-	•
STARTFIFO	•	•	•	•	•	•
STAT	•	•	•	•	•	•
STOLF	•	•	•	•	•	•
STOPFIFO	•	•	•	•	•	•
STOPRE	•	•	•	•	•	•
STOPREOF	•	•	•	•	•	•
STRING	•	•	•	•	•	•
STRINGFELD	•	•	•	•	•	•
STRINGIS	•	•	•	•	•	•
STRLEN	•	•	•	•	•	•
SUBSTR	•	•	•	•	•	•
SUPA	•	•	•	•	•	•
SVC	•	•	•	•	•	•
SYNFCT	•	•	•	•	•	•
SYNR	•	•	•	•	•	•
SYNRW	•	•	•	•	•	•
SYNW	•	•	•	•	•	•
T	•	•	•	•	•	•
TAN	•	•	•	•	•	•
TANG	○	○	○	○	○	○
TANGDEL	○	○	○	○	○	○
TANGOF	○	○	○	○	○	○
TANGON	○	○	○	○	○	○
TCA (828D: _TCA)	•	•	•	•	•	•
TCARR	•	•	•	•	•	•
TCI	•	•	•	•	•	•
TCOABS	•	•	•	•	•	•

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
● стандарт ○ опция - недоступно	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24х(5) CNC-SW G-Tech Экспорт (gse42)	SW26х(3) CNC-SW G-Tech Экспорт (gse62)	SW28х(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
TCOFR	●	●	●	●	●	●
TCOFRX	●	●	●	●	●	●
TCOFRY	●	●	●	●	●	●
TCOFRZ	●	●	●	●	●	●
THETA	-	-	-	-	-	-
TILT	-	-	-	-	-	-
TLIFT	○	○	○	○	○	○
TML	●	●	●	●	●	●
TMOF	●	●	●	●	●	●
TMON	●	●	●	●	●	●
TO	●	●	●	●	●	●
TOFF	●	●	●	●	●	●
TOFFL	●	●	●	●	●	●
TOFFOF	●	●	●	●	●	●
TOFFON	●	●	●	●	●	●
TOFFR	●	●	●	●	●	●
TOFRAME	●	●	●	●	●	●
TOFRAMEX	●	●	●	●	●	●
TOFRAMEY	●	●	●	●	●	●
TOFRAMEZ	●	●	●	●	●	●
TOLOWER	●	●	●	●	●	●
TOLENV	●	●	●	●	●	●
TOOLGNT	●	●	●	●	●	●
TOOLGT	●	●	●	●	●	●
TOROT	●	●	●	●	●	●
TOROTOF	●	●	●	●	●	●
TOROTX	●	●	●	●	●	●
TOROTY	●	●	●	●	●	●
TOROTZ	●	●	●	●	●	●
TOUPPER	●	●	●	●	●	●
TOWBCS	●	●	●	●	●	●
TOWKCS	●	●	●	●	●	●
TOWMCS	●	●	●	●	●	●
TOWSTD	●	●	●	●	●	●
TOWTCS	●	●	●	●	●	●
TOWWCS	●	●	●	●	●	●
TR	●	●	●	●	●	●
TRAANG	○	○	○	-	-	-

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
● стандарт ○ опция - недоступно						
TRACON	○	○	○	-	-	-
TRACYL	○	○	○	○	○	○
TRAFOOF	●	●	●	●	●	●
TRAFOON	-	-	-	-	-	-
TRAILOF	●	●	●	●	●	●
TRAILON	●	●	●	●	●	●
TRANS	●	●	●	●	●	●
Transmit	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-
TRUE	●	●	●	●	●	●
TRUNC	●	●	●	●	●	●
Tu	●	●	●	●	●	●
TURN	●	●	●	●	●	●
ULI	●	●	●	●	●	●
UNLOCK	●	●	●	●	●	●
UNTIL	●	●	●	●	●	●
UPATH	●	●	●	●	●	●
Var	●	●	●	●	●	●
VELOLIM	●	●	●	●	●	●
VELOLIMA	●	●	●	●	●	●
WAITC	●	●	●	●	●	●
WAITE	-	-	●	-	-	●
WAITENC	●	●	●	●	●	●
WAITM	-	-	●	-	-	●
WAITMC	-	-	●	-	-	●
WAITP	●	●	●	●	●	●
WAITS	●	●	●	●	●	●
WALCS0	●	●	●	●	●	●
WALCS1	●	●	●	●	●	●
WALCS2	●	●	●	●	●	●
WALCS3	●	●	●	●	●	●
WALCS4	●	●	●	●	●	●
WALCS5	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●
WALCS7	●	●	●	●	●	●
WALCS8	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●

21.2 Операторы: Доступность для SINUMERIK 828D

Оператор	SINUMERIK 828D					
	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)	SW24x(5) CNC-SW G-Tech Экспорт (gse42)	SW26x(3) CNC-SW G-Tech Экспорт (gse62)	SW28x(1) CNC-SW G-Tech Расшир. Эк- спорт (gse82)
• стандарт ○ опция - недоступно						
WALIMOF	•	•	•	•	•	•
WALIMON	•	•	•	•	•	•
WHEN	•	•	•	•	•	•
WHENEVER	•	•	•	•	•	•
WHILE	•	•	•	•	•	•
WORKPIECE	•	•	•	•	•	•
write	•	•	•	•	•	•
WRTPR	•	•	•	•	•	•
X	•	•	•	•	•	•
XOR	•	•	•	•	•	•
Y	•	•	•	•	•	•
Z	•	•	•	•	•	•

21.3 Текущий язык в HMI

Следующая таблица содержит все доступные на интерфейсе языки.

Текущий установленный язык может быть запрошен в программе обработки детали или в синхронных действиях через следующую системную переменную:

\$AN_LANGUAGE_ON_HMI = <значение>

<значение>	Язык	Краткое обозначение
1	Немецкий (Германия)	DEU
2	Французский	FRA
3	Английский (Соединенное королевство)	ENG
4	Испанский	ESP
6	Итальянский	ITA
7	Голландский	NLD
8	Китайский (упрощенный)	CHS
9	Шведский	SVE
18	Венгерский	HUN
19	Финский	FIN
28	Чешский	CSY
50	Португальский (бразильский)	PTB
53	Польский	PLK
55	Датский	DAN
57	Русский	RUS
68	Словацкий	SKY
72	Румынский	ROM
80	Китайский (традиционный)	CHT
85	Корейский	KOR
87	Японский	JPN
89	Турецкий	TRK

Примечание

Обновление \$AN_LANGUAGE_ON_HMI выполняется:

- после запуска системы.
- после сброса (Reset) ЧПУ и/или PLC.
- после переключения на другое ЧПУ в рамках M2N.
- после переключения языка на HMI.

A.1 Список сокращений

A	
A	Выход
ADI4	Analog Drive Interface for 4 Axes
AC	Адаптивное управление
ALM	Активный модуль питания
ARM	Асинхронный вращающийся двигатель
AS	Система автоматизации
ASCII	American Standard Code for Information Interchange: американский стандарт кода для передачи информации
ASIC	Application Specific Integrated Circuit: схема пользователя
ASUP	Асинхронная подпрограмма
AUXFU	Auxiliary Function: Вспомогательная функция
AWL	Список операторов
AWP	Программа пользователя

B	
BA	Режим работы
ГРР	Группа режимов работы (ГРР)
BCD	Binary Coded Decimals: закодированные двоичным кодом десятичные числа
BERO	Бесконтактный выключатель
BI	Binector Input (входной бинектор)
BICO	Binector Connector (бинектор-коннектор)
BIN	Binary Files: двоичные файлы
BIOS	Basic Input Output System
BKS	Базовая кинематическая система
BO	Binector Output (выходной бинектор)
OPI	Интерфейс пульта оператора

C	
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CC	Compile Cycle: компилируемые циклы
CEC	Cross Error Compensation
CI	Connector Input (входной коннектор)
CF-Card	Карта Compact Flash

C	
CNC	Computerized Numerical Control: компьютерное числовое программное управление
CO	Connector Output (выходной коннектор)
CoL	Сертификат лицензии
COM	Communication
CPA	Compiler Projecting Data: данные конфигурации компилятора
CRT	Cathode Ray Tube: кинескоп
CSB	Central Service Board: модуль PLC
CU	Control Unit
CP	Communication Processor
CPU	Central Processing Unit: центральный процессор
CR	Carriage Return
CTS	Clear To Send: сообщение о готовности к передаче для последовательных интерфейсов данных
CUTCOM	Cutter Radius Compensation: коррекция радиуса инструмента

D	
DAU	цифрово-аналоговый преобразователь
DB	Блок данных (PLC)
DBB	Байт блока данных (PLC)
DBD	Двойное слово блока данных (PLC)
DBW	Слово блока данных (PLC)
DBX	Бит блока данных (PLC)
DDE	Dynamic Data Exchange
DDS	Drive Data Set: Блок данных привода
DIN	Немецкий промышленный стандарт
DIO	Data Input/Output: индикация передачи данных
DIR	Directory: Директория
DLL	Dynamic Link Library
DO	Drive Object
DPM	Dual Port Memory
DPR	Dual-Port-RAM
DRAM	Динамическая память (не буферная)
DRF	Differential Resolver Function: функция дифференциального преобразования координат (маховичок)
Drive-Cliq	Drive Component Link with IQ
DRY	Dry Run: Подача пробного хода
DSB	Decoding Single Block: отдельный кадр декодирования
DSC	Dynamic Servo Control / Dynamic Stiffness Control
DW	Слово данных
DWORD:	двойное слово (сейчас 32 бита)

E	
E	Вход
EES	Execution from External Storage
I/O	Ввод/вывод
ENC	Encoder: датчик фактического значения
EFP	Простой периферийный модуль (PLC–I/O–модуль)
EGB	электростатически-чувствительные модули/детали
ЭМС	Электромагнитная совместимость (ЭМС)
EN	Европейский стандарт
ENC	Encoder: Датчик фактического значения
EnDat	Интерфейс датчика
EPROM	Erasable Programmable Read Only Memory: СППЗУ
Сетевые службы ePS	Службы для дистанционного обслуживания станков на базе Интернета
EQN	Обозначение типа абсолютного датчика с 2048 синусоидальными сигналами/оборот
ES	Engineering System (система технических разработок)
ESR	Расширенный останов и отвод
ETC	Клавиша расширения ">"; расширение панели программных клавиш в том же меню

F	
FB	Функциональный блок (PLC)
FC	Function Call: Функциональный блок (PLC)
FEPRM	Flash-EPROM: память для чтения и записи
FIFO	First in First Out: память, работающая без указания адреса, данные которой считываются в последовательности их сохранения
FIPO	Точный интерполятор
FPU	Floating Point Unit: блок вычислений (в режиме) с плавающей точкой
FRK	Коррекция радиуса фрезы
FST	Feed Stop: остановка подачи
FUP	Функциональные схемы FBD (метод программирования для PLC)
Микропрограммное обеспечение	Микропрограммное обеспечение

g	
GC	Global Control (PROFIBUS: широковещательная телеграмма)
GDIR	Глобальная память программ обработки деталей
GEO	Геометрия, например, гео-ось
GIA	Gear Interpolation Data: данные интерполяции редуктора
GND	Signal Ground
GP	Главная программа (PLC)
GS	Степень редуктора
GSD	Основной файл устройства для описания PROFIBUS Slave

g	
GSDML	Generic Station Description Markup Language: дескрипторный язык на базе XML для создания файла GSD
GUD	Global User Data: глобальные данные пользователя

h	
HEX	сокращение для шестнадцатеричного числа
HiFu	Вспомогательная функция
HLA	Гидравлический линейный привод
hmi	Human Machine Interface: интерфейс пользователя SINUMERIK
HSA	Привод главного движения
HW	Аппаратное обеспечение

I	
IBN	Ввод в эксплуатацию
IKA	интерполяционная компенсация
IM	Interface-Modul: модуль подключения
IMR	Interface-Modul Receive: модуль подключения для режима приема
IMS	Interface-Modul Send: модуль подключения для режима передачи
INC	Increment: размер шага
INI	Initializing Data: данные инициализации
IPO	Интерполятор
ISA	International Standard Architecture
ISO	International Standard Organization

J	
JOG	Jogging: отладочный режим

K	
K_v	Коэффициент усиления контура регулирования
K_p	Пропорциональное усиление
K_U	Передаточное число
KOP	Релейно-контактные схемы (LAD) (метод программирования для PLC)

l	
LAI	Logic Machine Axis Image: логический образ осей станка
LAN	Local Area Network
LCD	Liquid-Crystal Display: индикация на жидких кристаллах
Светодиод	Light Emitting Diode: светодиод
LF	Line Feed

I	
LMS	Система измерения положения
LR	Регулятор положения
LSB	Least Significant Bit: младший бит
LUD	Local User Data: данные пользователя (локальные)

M	
MAC	Media Access Control
MAIN	Main program: главная программа (OB1, PLC)
MB	Мегабайт
MCI	Motion Control Interface
MCIS	Motion–Control–Information–System
MCP	Machine Control Panel: Станочный пульт
MD	Машинные данные
MDA	Manual Data Automatic: ручной ввод
MDS	Motor Data Set: Блок данных двигателя
MELDW	Информационное слово
MCS	Система координат станка
MM	Модуль двигателя
MPF	Main Program File: Главная программа (ЧПУ)
MCP	Станочный пульт

N	
ЧПУ	Numerical Control: Числовое программное управление (ЧПУ) с подготовкой кадра, диапазоном перемещения и т. п.
NCU	Numerical Control Unit: аппаратный блок ЧПУ
NRK	Обозначение операционной системы ЧПУ
NST	Интерфейсный сигнал
NURBS	Non-Uniform Rational B-Spline
NV	Смещение нулевой точки
NX	Numerical Extension: модуль расширения осей

O	
OB	Организационный модуль в PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: панель оператора
OPI	Operation Panel Interface: подключение панели оператора
OPT	Options: опции
OLP	Optical Link Plug: разъем шины для световода
OSI	Open Systems Interconnection: взаимодействие открытых систем (сетевая архитектура)

P	
PAA	Образ процесса выходов
PAE	Образ процесса входов
PC	Personal Computer
PCIN	Имя программного обеспечения для обмена данными с СЧПУ
PCMCIA	Personal Computer Memory Card International Association: стандартизация карт памяти
PCU	PC Unit: PC-Box (блок ВУ)
PG	Программатор
PKE	Идентификатор параметра: часть PKW
PKW	Идентификатор параметра: значение (часть параметрирования PPO)
PLC	Programmable Logic Control: контроллер
PN	PROFINET
PNO	Организация пользователей PROFIBUS
po	POWER ON
POE	Структурная единица программы
Pos	Позиция/позиционирование
POSMO A	Positioning Motor Actuator: позиционирующий двигатель
POSMO CA	Positioning Motor Compact AC: модуль привода в сборе со встроенным силовым и управляющим модулем, а также блоком позиционирования и программной памятью; питание переменным током
POSMO CD	Positioning Motor Compact DC: как CA, но питание постоянным током
POSMO SI	Positioning Motor Servo Integrated: позиционирующий двигатель; питание постоянным током
PPO	Parameter Prozessdaten Objekt ; циклическая телеграмма данных при передаче с PROFIBUS-DP и профилем "Электрические приводы с регулируемой скоростью"
PPU	Panel Processing Unit (центральный аппаратный компонент панельной СЧПУ, например, SINUMERIK 828D)
Profibus	Process Field Bus: последовательная шина данных
PRT	Тест программы
PSW	Программное управляющее слово
PTP	Point to Point: точка-точка
PUD	Program Global User Data: глобальные программные переменные пользователя
PZD	Данные процесса: компонент данных процесса PPO

Q	
QFK	Компенсация квадратных ошибок

R	
RAM	Random Access Memory: память для чтения и записи
Ref	Функция подвода в референтную точку
REPOS	Функция репозиционирования

R	
RISC	Reduced Instruction Set Computer: тип процессора с небольшим набором команд и быстрым прохождением команд
ROV	Rapid Override: входная коррекция
RP	R-параметр, предопределенная переменная пользователя
RPA	R-Parameter Active: Область памяти в ЧПУ для номеров R-параметров
RPY	Roll Pitch Yaw: тип вращения системы координат
RTL	Rapid Traverse Linear Interpolation: линейная интерполяция при движении ускоренного хода
RTS	Request To Send: включение блока передачи, сигнал управления от последовательных интерфейсов данных
RTCP	Real Time Control Protocol

S	
SA	Синхронное действие
SBC	Safe Break Control: безопасное управление торможением
SBL	Single Block: отдельный кадр
SBR	Subroutine: подпрограмма (PLC)
SD	Установочные данные
SDB	Системный блок данных
SEA	Setting Data Active: обозначение (тип файла) для установочных данных
SERUPRO	Search-Run by Program Test: Поиск кадра через программный тест
SFB	Системный функциональный блок
SFC	System Function Call
SGE	Безопасно-ориентированный вход
SGA	Безопасно-ориентированный выход
SH	Безопасный останов
SIM	Single in Line Module
SK	Программная клавиша
SKP	Skip: функция для пропуска кадров программы обработки деталей
SLM	Синхронный линейный двигатель
SM	Шаговый электродвигатель
SMC	Sensor Module Cabinet Mounted
SME	Sensor Module Externally Mounted
SMI	Sensor Module Integrated
SPF	Main Program File: подпрограмма (ЧПУ)
SPS	Контроллер = PLC
SRAM	Статическая память (буферная)
SRK	Коррекция радиуса резцов
SRM	Синхронный вращающийся двигатель
SSFK	Компенсация погрешности ходового винта
SSI	Serial Synchron Interface: последовательный синхронный интерфейс
SSL	Поиск кадра
STW	Управляющее слово

S	
SUG	Окружная скорость круга
SW	ПО
SYF	System Files: системные файлы
SYNACT	Synchronized Action: Синхронное действие

T	
TB	Terminal Board (SINAMICS)
TCP	Tool Center Point: острие инструмента
TCP/IP	Transport Control Protocol / Internet Protocol
TCU	Thin Client Unit
TEA	Testing Data Active: идентификатор для машинных данных
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset: Коррекция инструмента
TOA	Tool Offset Active: обозначение (тип файла) для коррекций инструмента
Transmit	Transform Milling into Turning: трансформация координат для фрезерной обработки на токарном станке
TTL	транзисторные логические схемы со связями на транзисторах (тип интерфейса)
TZ	Технологический цикл

U	
UFR	User Frame: Смещение нулевой точки
UP	Подпрограмма
USB	Universal Serial Bus
USV	ИБП

V	
VDI	Внутренний коммуникационный интерфейс между ЧПУ и PLC
VDI	Союз немецких инженеров
VDE	Союз немецких электротехников
VI	Voltage Input
VO	Voltage Output
VSA	Привод подачи

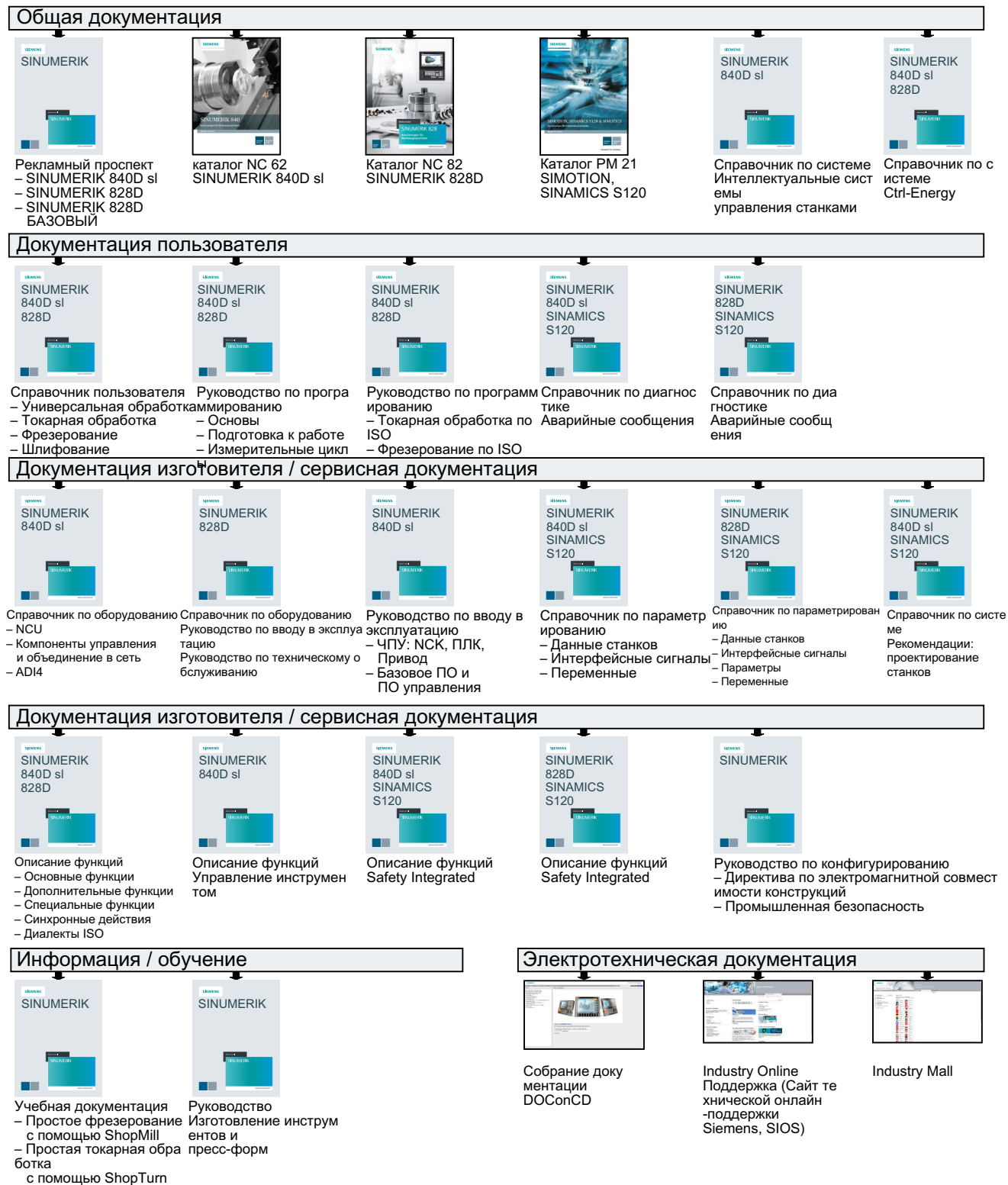
W	
WAB	Функция мягкого подвода и отвода (SAR)
WCS	Система координат детали
WKZ	Инструмент
WLK	Коррекция длины инструмента
WOP	программирование у производственного оборудования

W	
WPD	Work Piece Directory: директория детали
WRK	Коррекция радиуса инструмента (КРИ)
WZ	Инструмент
WZK	Коррекция инструмента
WZV	Управление инструментом
WZW	Смена инструмента

X	
XML	Extensible Markup Language

Z	
ZOA	Zero Offset Active: идентификатор для смещений нулевой точки
ZSW	Слово состояния (привода)

A.2 Обзор документации



Толковый словарь

CNC

См. -> ЧПУ

Computerized Numerical Control: охватывает компоненты → NC, → PLC, HMI, → COM.

CNC

См. -> ЧПУ

Computerized Numerical Control: охватывает компоненты → NC, → PLC, HMI, → COM.

COM

Компонент СЧПУ для осуществления и координации коммуникации.

CPU

Central Processor Unit, см. -> Контроллер

С-сплайн

С-сплайн это самый известный и наиболее распространенный сплайн. Переходы на опорных точках являются стабильными по касательной и кривизне. Используются полиномы 3-го порядка.

DRF

Differential Resolver Function: функция ЧПУ, создающая вместе в электронным маховичком инкрементальное смещение нулевой точки в автоматическом режиме.

HIGHSTEP

Система возможностей программирования для -> PLC системы AS300/AS400.

HW-Konfig

ПО SIMATIC S7 для конфигурирования и параметрирования аппаратных компонентов в рамках проекта S7.

JOG

Режим работы СЧПУ (отладочный режим): в режиме работы Jog может осуществляться отладка станка. Отдельные оси и шпиндели через клавиши направления могут

перемещаться в периодическом режиме. Прочими функциями режима работы Jog являются -> Реферирование, -> Repos и -> Preset (установка фактического значения).

KU

Передаточное отношение

KV

Коэффициент усиления контура, величина техники автоматического регулирования регулирующего контура

Look Ahead

С помощью функции **Look Ahead** посредством "опережения" на параметрируемое количество кадров перемещения достигается оптимальная скорость обработки.

MDA

Режим работы СЧПУ: Manual Data Automatic. В режиме MDA отдельные программные кадры или последовательности кадров могут вводиться без ссылки на главную или подпрограмму, и после этого сразу же выполняться через клавишу NC-Start.

NRK

Numeric Robotic Kernel (операционная система → NC)

NURBS

Управление движением и траекторная интерполяция со стороны СЧПУ осуществляются на основе NURBS (**N**on **U**niform **R**ational **B**-**S**plines). Тем самым внутри СЧПУ для всех интерполяций доступен единый метод.

OEM

Для изготовителей станка, которые хотят создавать свои собственные интерфейсы пользователя или внедрять специфические технологические функции в СЧПУ, предусмотрены свободные зоны для индивидуальных решений (приложения OEM).

PLC

Programmable Logic Control: → Контроллер. Компонент → ЧПУ: адаптивное управления для обработки логического контроля станка.

R-параметр

R-параметр, может устанавливаться и запрашиваться программистом → Программы обработки деталей для любых целей в программе.

V.24

Последовательный интерфейс для ввода/вывода данных. Через этот интерфейс могут загружаться и сохраняться управляющие программы, а также данные изготовителя и пользователя.

WinSCP

WinSCP это бесплатная открытая программа (Open Source) для Windows для передачи файлов.

Абсолютный размер

Указание цели движения оси через размер, относящийся к нулевой точке действующей в данный момент системы координат. См. также -> Составной размер.

Автоматика

Режим работы СЧПУ (режим последовательных кадров по DIN): Режим работы для систем ЧПУ, в котором включается и последовательно выполняется -> Программа обработки деталей.

Адрес

Адрес это обозначение для определенных операндов или области операндов, к примеру, вход, выход и т.д.

Адрес оси

См. → Имя оси

Архивация

Выгрузка данных и/или директорий на **внешнее** ЗУ.

Асинхронная подпрограмма

Программа, которая может запускаться асинхронно (независимо) от актуального состояния программы через сигнал прерываний (к примеру, сигнал "быстрый вход ЧПУ").

Базовая кинематическая система

Декартова система координат через трансформацию отображается на систему координат станка.

В -> Программе обработки деталей программист использует имена осей базовой кинематической системы. Она существует, если нет активной -> Трансформации, параллельно -> Системе координат станка. Отличие состоит в → именах осей.

Базовая ось

Ось, заданное и фактическое значение которой используются для вычисления значения компенсации.

Блок

Блоком называются все файлы, которые необходимы для создания и обработки программы.

Блок ТОА

Каждая → Область ТОА может содержать несколько блоков ТОА. Число макс. возможных блоков ТОА ограничивается через макс. число активных → Каналов. Один блок ТОА включает один блок данных инструмента и один блок данных магазина. Дополнительно может быть включен еще один блок данных инструментального суппорта (опция).

Блок данных

1. Блок данных -> PLC, к которому могут обращаться -> Программы HIGHSTEP.
2. Единица данных -> ЧПУ: Блоки данных содержат определения для глобальных данных пользователя. Данные могут подвергаться прямой инициализации при определении.

Буферная батарея

Буферная батарея обеспечивает энергонезависимое сохранение -> Программы пользователя в -> CPU и постоянное сохранение определенных областей данных и меток, таймеров и счетчиков.

Быстрые цифровые входы/выходы

Через цифровые входы, к примеру, могут запускаться быстрые программы ЧПУ (обработчики прерываний). Через цифровые выходы ЧПУ могут запускаться быстрые, управляемые программой функции комбинационной логики.

Быстрый отвод от контура

При возникновении прерывания через программу обработки ЧПУ может быть запущено движение, которое позволяет быстро отвести инструмент от обрабатываемого в данный момент контура детали. Дополнительно могут быть спараметрированы угол отвода и значение пути. После быстрого отвода дополнительно может быть выполнен обработчик прерываний.

Ведомая ось

Ведомая ось это → Ось Gantry, заданная позиция которой всегда является производной от движения перемещения → Ведущей оси и тем самым она перемещается синхронно. С точки зрения оператора и программиста ведомая ось "отсутствует".

Ведущая ось

Ведущая ось это → Ось Gantry, которая, с точки зрения оператора и программиста, присутствует на станке и поэтому может управляться как обычная ось ЧПУ.

Винтовая интерполяция

Винтовая интерполяция особенно подходит для простого изготовления внутренних или наружных резьб с помощью профильных фрез и для фрезерования смазочных канавок.

При этом винтовая линия составляется из двух движений:

- Круговое движение в плоскости
- Линейное движение вертикально к этой плоскости.

Внешнее смещение нулевой точки

Заданное с → PLC смещение нулевой точки.

Вращение

Компонент → Фрейма, который определяет поворот системы координат на определенный угол.

Вспомогательные функции

С помощью вспомогательных функции в → Программах обработки деталей на → PLC могут передаваться → Параметры, которые запускают там определенные изготовителем станка реакции.

Вспомогательный кадр

Вводимый "N" кадр с информацией по рабочей операции, к примеру, с указанием позиции.

Высокоуровневый язык программирования ЧПУ

Высокоуровневый язык служит для записи программ для ЧПУ, → синхронных действий и → циклов. Он предлагает: Управляющие структуры, → Определенные пользователем переменные, → Системные переменные, → Технику макросов.

Геометрическая ось

Гео-оси образуют 2- или 3-мерную → Систему координат детали, в которой в → Программах обработки деталей программируется геометрия детали.

Геометрия

Описание -> Детали в -> Системе координат детали.

Главная программа

Наименование "Главная программа" относится к тому времени, когда существовало жесткое разделение программы обработки детали на главную и → Подпрограммы. Такого жесткого разделения в современном языке ЧПУ SINUMERIK больше нет. В принципе, любая программа обработки детали может быть выбрана и запущена в канале. После она выполняется на → Программном уровне 0 (уровень главной программы). В главной программе другие программы обработки детали или → Циклы могут вызываться как подпрограммы

Главный кадр

Вводимый через ":" кадр, содержащий все данные, необходимые для запуска процесса работы в → Программе обработки деталей.

Граница точного останова

При достижении всеми траекторными осями их границы точного останова, СЧПУ ведет себя так, как будто оно точно достигло заданной точки. Осуществляется переключение кадра -> Программы обработки деталей.

Группа режимов работы (ГРР)

Технологически связанные оси и шпиндели могут быть объединены в группу режимов работы (ГРР). Оси/шпиндели одной ГРР могут управляться из одного или нескольких → Каналов. С каналами ГРР всегда согласован один и тот же -> Режим работы.

Деталь

Создаваемая/обрабатываемая на станке часть.

Диагностика

1. Область управления СЧПУ
2. СЧПУ имеет как программу самодиагностики, так и средства тестирования для сервисных целей: индикации состояния, ошибок и сервисные индикации.

Диапазон перемещения

Максимальный допустимый диапазон перемещения для линейных осей составляет ± 9 декад. Абсолютное значение зависит от выбранной дискретности ввода и управления положением и системы единиц (дюймовая или метрическая).

Дюймовая система единиц

Система единиц, определяющая расстояния в дюймах и их долях.

Заготовка

Часть, с которой начинается обработка детали.

Загрузка

Загрузка системной программы после Power On.

Защищенное пространство

Трехмерное пространство внутри → Рабочего пространства, куда не должно заходить острие инструмента.

Значение компенсации

Разница между измеренной датчиком позицией оси и желаемой, запрограммированной позицией оси.

Идентификатор

Слова по DIN 66025 через идентификаторы (имена) для переменных (R-переменные, системные переменные, переменные пользователя), для подпрограмм, для кодовых слов и слов расширяются несколькими буквами адреса. Значение этих расширений идентично словам в структуре кадра. Идентификаторы должны быть однозначными. Один и тот же идентификатор не может использоваться для различных объектов.

Изгиб

Изгиб k контура это обратная величина радиуса r прилегающей окружности в точке контура ($k = 1/r$).

Имя оси

Для однозначной идентификации все оси канала и → оси станка СЧПУ должны получить оригинальные для канала или СЧПУ имена. → геометрические оси обозначаются с X, Y, Z. Вращающиеся вокруг гео-осей → круговые оси обозначаются с A, B, C.

Инструмент

Действующий компонент станка, отвечающая за обработку (к примеру, токарный резец, фреза, сверло, луч лазера).

Интерполятор

Логическая единица → NC, которая после указания заданного конечного положения в программе обработки деталей определяет промежуточные значения для движений, проходимых отдельными осями.

Интерполяционная компенсация

С помощью интерполяционных компенсаций, таких как → компенсация погрешности шага винта, прогиба, угловатости и температуры, компенсируются механические ошибки станка.

Интерфейс управления

Интерфейс управления (BOF) это среда индикации СЧПУ в форме дисплея. Он образуется горизонтальными и вертикальными программными клавишами.

Кадр программы обработки детали

Часть → Программы обработки деталей, ограниченная переходом на новую строку. Различаются → Главные кадры и → Вспомогательные кадры.

Канал

Свойством канала является его способность выполнять → Программу обработки детали, независимо от других каналов. Канал осуществляет эксклюзивное управление согласованными с ним осями и шпинделями. Процессы программы обработки детали различных каналов могут координироваться через → Синхронизацию.

Канал обработки

Благодаря канальной структуре через параллельные процессы движения может сокращаться вспомогательное время, к примеру, перемещения портала загрузки синхронно с обработкой. Канал ЧПУ при этом рассматривается как своя СЧПУ с декодированием, подготовкой кадра и интерполяцией.

Кодовые слова

Слова с фиксированным написанием, которые имеют в языке программирования для - > Программ обработки деталей определенное значение.

Кодовый переключатель

Кодовый переключатель на -> Станочном пульте имеет 4 позиции, которым операционной системой СЧПУ присвоены функции. Кроме этого, к кодовому переключателю относятся три ключа разного цвета, которые могут выниматься в указанных позициях.

Компенсация квадрантных ошибок

Ошибки контура на квадрантных переходах, возникающие из-за переменных соотношений трения на направляющих, могут быть практически устранены благодаря компенсации квадрантных ошибок. Параметрирование компенсации квадрантных ошибок осуществляется через круговой тест.

Компенсация люфта

Компенсация механического люфта станка, к примеру, обратного люфта у шариковинтовых пар. Для каждой оси компенсация люфта может вводиться отдельно.

Компенсация погрешности ходового винта

Компенсация механических неточностей участвующей в подаче шариковинтовой пары через СЧПУ на основе имеющихся измеренных величин отклонения.

Контроллер

Контроллеры (SPS) это электронные управления, функция которых сохранена как программа в устройстве управления. Таким образом, конструкция и проводка прибора не зависят от функции СЧПУ. Контроллер имеет структуру ВУ; он состоит из CPU (центральный модуль) с памятью, модулей ввода/вывода и внутренней шинной системы. Периферийные устройства и язык программирования соответствуют требованиям техники автоматического управления.

Контроль контура

В качестве меры точности контура контролируется погрешность запаздывания в пределах определенного диапазона допуска. Недопустимо высокая погрешность запаздывания может возникнуть, к примеру, из-за перегрузки привода. В этом случае сигнализируется ошибка и оси останавливаются.

Контур

Очертания -> Детали

Контур готовой детали

Контур детали после завершения обработки. См. -> Заготовка.

Контур детали

Заданный контур создаваемой/обрабатываемой -> Детали.

Коррекция инструмента

Учет размеров инструмента при вычислении траектории.

Коррекция радиуса инструмента

Для прямого программирования желаемого -> Контура детали СЧПУ, с учетом радиуса используемого инструмента, должна перемещаться по эквидистантной траектории к запрограммированному контуру (G41/G42).

Коррекция радиуса резцов

При программировании контура за основу берется острый инструмент. Так как это не может быть реализовано на практике, то радиус изгиба используемого инструмента сообщается СЧПУ и учитывается ей. При этом центр изгиба, смещенный на радиус изгиба, ведется эквидистантно вокруг контура.

Круговая интерполяция

-> Инструмент должен двигаться между установленными точками контура с заданной подачей по кругу, обрабатывая при этом деталь.

Круговая ось

Круговые оси вызывают поворот детали или инструмента в заданное угловое положение.

Линейная интерполяция

Инструмент перемещается по прямой к заданной точке, обрабатывая при этом деталь.

Линейная ось

Линейная ось это ось, которая, в отличие от круговой оси, описывает прямую.

Масса

Массой называется совокупность всех связанных между собой пассивных частей оборудования, которые и в случае ошибки не принимают опасного контактного напряжения.

Масштабирование

Компонент -> Фрейма, вызывающий специфическое для осей изменение масштаба.

Метрическая измерительная система

Нормированная система единиц: для длин, к примеру, мм (миллиметр), м (метр).

Метрическое и дюймовое указание размера

В программе обработки значения позиций и шага могут быть запрограммированы в дюймах. Независимо от программируемого указания размера (G70/G71) СЧПУ настраивается на основную систему.

Наклонная обработка

Сверлильные и фрезеровальные обработки на деталях, находящихся не в плоскости координат станка, могут осуществляться с поддержкой функции "наклонная обработка".

Нарезание внутренней резьбы без компенсирующего патрона

С помощью этой функции можно нарезать внутреннюю резьбу без компенсационного патрона. Благодаря интерполирующему перемещению шпинделя в качестве круговой оси и оси сверления резьба нарезается точно до конечной глубины сверления, к примеру, глухая резьба (условие: осевой режим шпинделя).

Нулевая точка детали

Нулевая точка детали образует исходную точку для → Системы координат детали. Она определяется через расстояния до → Нулевой точки станка.

Нулевая точка станка

Фиксированная точка станка, к которой могут быть привязаны все (зависимые) системы измерения.

Область ТОА

Область ТОА включает в себя все данные инструментов и магазинов. Стандартно эта область касательно диапазона действия совпадает с областью → Канал. Но через машинные данные может быть установлено, что несколько каналов используют один → Блок ТОА, таким образом, этим каналам доступны общие данные управления инструментом.

Обработчик прерываний

Обработчики прерываний это специальные → Подпрограммы, которые могут запускаться событиями (внешними сигналами) из процесса обработки. Находящийся в обработке кадр программы обработки деталей отменяется, позиция прерывания осей автоматически сохраняется.

Обратная по времени подача

Вместо скорости подачи для движения оси может быть запрограммировано и время, необходимо для хода траектории одного кадра (G93).

Ограничение рабочего поля

С помощью ограничения рабочего поля в дополнение к конечным выключателям можно ограничить диапазон перемещения осей. На ось возможна пара значений для описания защищенного рабочего пространства.

Оперативная память

Оперативная память это память RAM в -> CPU, к которой обращается процессор при обработке программы пользователя.

Определение переменных

Определение переменных включает в себя определение типа данных и имени переменной. С помощью имени переменной может осуществляться обращение к значению переменной.

Определенная пользователем переменная

Пользователь для любого использования в -> Программе обработки детали или блоке данных (глобальные данные пользователя) может согласовывать определенные пользователем переменные. Определение включает указание типа данных и имя переменной. См. -> Системная переменная.

Ориентированный останов шпинделя

Останавливает шпиндель изделия в заданном угловом положении, чтобы, к примеру, осуществить дополнительную обработку в определенном месте.

Оси

Оси ЧПУ, в соответствии с объемом их функций, подразделяются следующим образом:

- Оси: интерполирующие траекторные оси
- Вспомогательные оси: не интерполирующие оси подачи и позиционирующие оси со специфической для оси подачи. Вспомогательные оси не участвуют в самом процессе обработки, к примеру, подача инструмента, инструментальный магазин.

Оси станка

Физически существующие оси станка.

Ось C

Ось, вокруг которой осуществляется управляемое движение вращения и позиционирование с помощью шпинделя изделия.

Ось закругления

Оси округления вызывают поворот детали или инструмента в угловое положение, соответствующее делительному растру. При достижении растра ось округления находится "в позиции".

Ось компенсации

Ось, заданное и фактическое значение которой модифицируется через значение компенсации.

Отражение

При отражении меняются знаки значений координат контура относительно оси. Отражение может осуществляться одновременно относительно нескольких осей.

Ошибки

Все -> Сообщения и ошибки показываются на пульте оператора текстом с датой, временем и соответствующим символом для критерия стирания. Индикация осуществляется отдельно по ошибкам и сообщениям.

1. Ошибки и сообщения в программе обработки детали
Ошибки и сообщения могут индизироваться текстом непосредственно из программы обработки детали.
2. Ошибки и сообщения PLC
Ошибки и сообщения могут индизироваться текстом непосредственно из программы PLC. Дополнительных пакетов функциональных блоков для этого не требуется.

Память загрузки

Память загрузки у CPU 314 -> SPS идентична -> Оперативной памяти.

Память коррекций

Область данных в СЧПУ, в которой сохраняются данные коррекции инструмента.

Память пользователя

Все программы и данные, как то программы обработки деталей, подпрограммы, комментарии, коррекции инструмента, смещения нулевой точки/фреймы, а также каналные и программные данные пользователя могут сохраняться в общей памяти ЧПУ пользователя.

Периферийный модуль

Периферийные модули создают соединение между CPU и процессом.

Периферийными модулями являются:

- -> Цифровые модули ввода/вывода
- -> Аналоговые модули ввода/вывода
- → Модули симулятора

Подача по траектории

Подача по траектории действует на -> Траекторные оси. Она представляет собой геометрическую сумму подача участвующих -> Геометрических осей.

Подвод к фиксированной точке

Станки могут осуществлять определенный подвод к фиксированным точкам, как то точка смены инструмента, точка загрузки, точка смены палет и т.п. Координаты этих точек зафиксированы в СЧПУ. СЧПУ перемещает соответствующие оси, если возможно, -> Ускоренным ходом.

Подпрограмма

Наименование "Подпрограмма" относится к тому времени, когда существовало жесткое разделение программы обработки детали → на Главную и подпрограммы. Такого жесткого разделения в сегодняшнем языке ЧПУ SINUMERIK больше нет. В принципе, любая программа обработки детали или любой → Цикл могут быть вызваны в другой программе обработки детали как подпрограмма. После она выполняется на следующем → Программном уровне (x+1) (уровень подпрограмма (x+1)).

Позиционирующая ось

Ось, выполняющая вспомогательное движение на станке. (к примеру, инструментальный магазин, транспортировка палет). Позиционирующие оси это оси, которые не выполняют интерполяции с -> Траекторными осями.

Поиск кадра

Для тестирования программ обработки деталей или после отмены обработки с помощью функции "Поиск кадра" может быть выбрано любое место в программе обработки деталей, с которого обработка должна быть запущена или продолжена.

Полиномиальная интерполяция

С помощью полиномиальной интерполяции могут создаваться различные ходы кривой, как то **прямолинейная, параболическая, степенная функции** (SINUMERIK 840D sl).

Полярные координаты

Система координат, определяющая положение точки в плоскости через ее расстояние до нулевой точки и угол, который образует вектор радиуса с определенной осью.

Предельное число оборотов

Максимальное/минимальное число оборотов (шпинделя): Через задачу машинных данных, -> PLC или -> Установочных данных максимальное число оборотов шпинделя может быть ограничено.

Предсовпадение

Смена кадра уже при приближении ходом траектории к конечной позиции на заданную дельту.

Предуправление, динамическое

Неточности → Контура, обусловленные ошибками запаздывания, могут быть практически полностью устранены через динамическое, зависимое от ускорения предупреждение. Благодаря этому, даже при высоких → Скоростях движения по траектории, получается отличная точность обработки. Включение и выключение предупреждения возможно спец. для оси через → Программу обработки детали.

Привод

Привод это модуль ЧПУ, выполняющий управление числом оборотов и моментами на основе данных с ЧПУ.

Программа обработки детали

Последовательность операторов на ЧПУ, которые вместе способствуют созданию определенной -> Детали. Также и осуществление определенной обработки на имеющейся -> Заготовке.

Программа передачи данных PCIN

PCIN это вспомогательная программа для отправки и получения данных пользователя ЧПУ через последовательный интерфейс, к примеру, программ обработки деталей, коррекции инструмента и т.п. Программа PCIN может работать под MS-DOS на стандартных промышленных ПК.

Программа пользователя

Программы пользователя для систем автоматизации S7-300 создаются с помощью языка программирования STEP 7. Программа пользователя имеет модульную структуру и состоит из отдельных блоков.

Основными типами блоков являются:

- Блоки кода
Эти блоки содержат команды STEP 7.
- Блоки данных
Эти блоки содержат постоянные и переменные для программы STEP 7.

Программирование PLC

PLC программируется с помощью ПО **STEP 7**. ПО программирования STEP 7 основывается на стандартной операционной системе **WINDOWS** и включает в себя функции программирования STEP 5 с новейшими разработками.

Программируемое ограничение рабочего поля

Ограничение зоны движения инструмента до определенной через запрограммированные ограничения зоны.

Программируемые фреймы

С помощью программируемых -> Фреймов в ходе выполнения программы обработки деталей может осуществляться динамическое определение новых исходных точек системы координат. Различается абсолютное определение на основе нового фрейма и аддитивное определение с ссылкой на существующую исходную точку.

Программная клавиша

Клавиша, надпись которой представляется в поле на дисплее и динамически изменяется в зависимости от актуальной ситуации управления. Функциональные клавиши со свободным присвоением функции (программные клавиши) согласуются с определенными на программном уровне функциями.

Программная память PLC

SINUMERIK 840D sl: В памяти пользователя PLC сохраняются программа электроавтоматики и данные пользователя вместе с главной программой PLC.

Программные конечные выключатели

Программные конечные выключатели ограничивают диапазон перемещения оси и предотвращают наезд салазок на аппаратные конечные выключатели. На ось могут задаваться 2 пары значений, которые могут активироваться отдельно через -> PLC.

Программный блок

Программные блоки содержат главные и подпрограммы -> Программ обработки детали.

Программный код

Символы и последовательность символов, имеющие в языке программирования для → Программ обработки деталей определенное значение.

Программный уровень

Запущенная в канале программа обработки детали выполняется как → Главная программа на программном уровне 0 (уровень главной программы). Любая вызванная в главной программе программа обработки детали выполняется как → Подпрограмма на собственном программном уровне 1 ... n.

Промежуточные кадры

Движения перемещения с выбранной → Коррекцией инструмента (G41/G42) могут прерываться ограниченным количеством промежуточных кадров (кадры без осевых движений в плоскости коррекции), при этом коррекция инструмента может вычисляться корректно. Допустимое количество промежуточных кадров, предварительно считываемых СЧПУ, может устанавливаться через системные параметры.

Процентовка

Ручная или программируемая возможность вмешательства, позволяющая оператору осуществлять наложение запрограммированных подач или числа оборотов, чтобы согласовать их с определенной деталью или материалом.

Процентовка подачи

На запрограммированную скорость накладывается актуальная установка скорости через → Станочный пульт или от → PLC (0-200%). Скорость подачи может дополнительно корректироваться в программе обработки через программируемый процентный коэффициент (1-200%).

Рабочее пространство

Трехмерное пространство, в которое может входить острие инструмента на основании конструкции станка. См. → Защищенное пространство.

Размер шага

Указание длин пути перемещения через количество инкрементов (размер шага). Количество инкрементов может сохраняться как → Установочные данные или выбираться через клавиши с соответствующими надписями 10, 100, 1000, 10000.

Редактор

Редактор обеспечивает создание, изменение, дополнение, соединение и вставку программ/текстов/кадров программы.

Режим работы

Концепция работы СЧПУ SINUMERIK. Определены режимы работы -> Jog, -> MDA, -> Автоматический.

Режим управления траекторией

Целью режима управления траекторией является недопущение сильного торможения → Траекторных осей на границах кадров программы обработки деталей и переход в следующий кадр по возможности с равномерной скоростью движения по траектории.

Референтная точка

Точка станка, к которой относится система измерения -> Осей станка.

Сеть

Сеть это соединение нескольких S7-300 и других оконечных устройств, к примеру, программаторов, через -> Соединительный кабель. Через сеть осуществляется обмен данными между подключенными устройствами.

Синхронизация

Операторы в -> Программах обработки деталей для координации процессов в различных -> Каналах в определенных местах обработки.

Синхронные действия

1. Вывод вспомогательной функции
При обработке детали из программы ЧПУ могут выводиться технологические функции (-> Вспомогательные функции) на PLC. Через эти вспомогательные функции осуществляется, к примеру, управление дополнительными устройствами станка, к примеру, пинолью, захватом, зажимным патроном и т.п.
2. Быстрый вывод вспомогательных функций
Для критических по времени функций переключения могут быть минимизированы времена квитирования для -> Вспомогательных функций и исключаются ненужные точки остановки в процессе обработки.

Синхронные оси

Для хода синхронных осей требуется то же время, что и для хода геометрических осей по траектории.

Система координат

См. -> Система координат станка, -> Система координат детали

Система координат детали

Исходная точка системы координат детали находится в
-> Нулевой точке детали. При программировании в системе координат детали размеры и направления относятся к этой системе.

Система координат станка

Система координат, относящаяся к осям станка.

Системная память

Системная память это память в CPU, в которой сохраняются следующие данные:

- данные, необходимые для операционной системы
- Операнды таймеры, счетчики, меркеры

Системная переменная

Переменная, существующая без вмешательства программиста -> Программы обработки деталей. Она определена через тип данных и имя переменной, которое вводится символом \$. См. -> Определенные пользователем переменные.

Скорость движения по траектории

Максимальная программируемая скорость движения по траектории зависит от дискретности ввода. При разрешении, к примеру, 0,1 мм максимальная программируемая траекторная скорость движения по траектории составляет 1000 м/мин.

Скорость передачи данных

Скорость при передаче данных (бит/сек).

Слово данных

Единица данных размером в два байта внутри -> Блока данных.

Смещение нулевой точки

Указание новой исходной точки для системы координат через отношение к существующей нулевой точке и -> Фрейм.

1. Устанавливаемое
Доступно проектируемое количество устанавливаемых смещений нулевой точки для каждой оси ЧПУ. Включаемые через команды G смещения действуют альтернативно.
2. Внешнее
В дополнение ко всем смещениям, определяющим положение нулевой точки детали, возможно наложение внешнего смещения нулевой точки через маховичок (смещение DRF) или с PLC.
3. Программируемое
С помощью оператора TRANS для всех траекторных и позиционирующих осей могут программироваться смещения нулевой точки.

Сообщения

Все запрограммированные в программе обработки детали сообщения и определенные системой -> Ошибки показываются на пульте оператора текстом с датой, временем и соответствующим символом для критерия стирания. Индикация осуществляется раздельно по ошибкам и сообщениям.

Составной размер

Также инкрементальный размер: указание цели движения оси через проходимые участки пути и направление относительно уже достигнутой точки. См. -> Абсолютный размер.

Сплайн-интерполяция

С помощью сплайн-интерполяции СЧПУ может создать ровный ход кривой из малого количества заданных опорных точек заданного контура.

Стандартные циклы

Для часто повторяющихся задач обработки имеются стандартные циклы:

- Для сверлильной/фрезерной технологии
- Для токарной технологии

В области управления "Программа" в меню "Поддержка циклов" перечислены доступные циклы. После выбора желаемого цикла обработки необходимые параметры для присвоения значений показываются текстом.

Станочный пульт

Пульт оператора станка с элементами управления, клавишами, поворотными выключателями и т.д. и простыми элементами индикации, как то LED. Они служат для непосредственного управления станком через PLC.

Стирание до первичного состояния

При стирании до первичного состояния стираются следующие области памяти -> CPU:

- --> Оперативная память
- область записи/чтения -> Памяти загрузки
- -> Системная память
- -> Резервная память

Таблица компенсаций

Таблица опорных точек. Она дает значения компенсации оси компенсации для выбранных позиций базовой оси.

Текстовый редактор

См. -> Редактор

Техника макросов

Комбинация нескольких операторов под одним идентификатором. Идентификатор представляет в программе набор связанных операторов.

Точный останов

При запрограммированном операторе точного останова осуществляться точный и при необходимости очень медленный подвод к указанной в кадре позиции. Для уменьшения времени сближения для ускоренного хода и подачи определяются
-> Границы точного останова.

Траекторная ось

Траекторными осями являются все оси обработки -> Канала, управляемые -> Интерполятором таким образом, что они одновременно запускаются, ускоряются, останавливаются и достигают конечной точки.

Трансформация

Аддитивное или абсолютное смещение нулевой точки оси.

Управление программой обработки детали

Управление программой обработки деталей может быть организовано по -> Деталям. Количество управляемых программ и данных зависит от объема памяти пользователя. Каждому файлу (программе и данным) может быть присвоено имя из макс. 24-ти алфавитно-цифровых символов.

Управление скоростью

Для достижения приемлемой скорости перемещения при движениях перемещения на очень маленькие значения в кадре может быть установлена опережающая обработка на несколько кадров (-> Look Ahead).

Ускорение с ограничением рывка

Для получения оптимальной характеристики ускорения на станке при одновременном щадящем воздействии на механику в программе обработки можно переключаться между скачкообразным ускорением и постоянным (плавным) ускорением.

Ускоренный ход

Самая быстрая скорость перемещения оси. Она используется, к примеру, для подвода инструмента из состояния покоя к -> Контуру детали или отвода от контура детали. Скорость ускоренного хода устанавливается спец. для станка через машинные данные.

Установочные данные

Данные, сообщающие свойства станка способом, определенным через системное ПО, на ЧПУ.

Фиксированная точка станка

Однозначно определенная через станок точка, к примеру, референтная точка станка.

Фрейм

Фрейм представляет собой правило вычисления, переводящее одну декартову систему координат в другую декартову систему координат. Фрейм содержит компоненты -> Смещение нулевой точки, -> Вращение, -> Масштабирование, -> Отражение.

Функции безопасности

СЧПУ имеет постоянно активные контроли, которые заранее распознают сбои в -> ЧПУ, контроллере (-> PLC) и на станке, что практически исключает повреждения детали, инструмента или станка. В случае сбоя процесс обработки прерывается и приводы останавливаются, причина сбоя сохраняется и показывается как ошибка. Одновременно на PLC сообщается, что имеет место ошибка ЧПУ.

Циклы

Защищенные подпрограммы для исполнения повторяющегося процесса обработки на -> Детали.

ЧПУ

Компонент → ЧПУ, который выполняет → программы обработки деталей и координирует процессы движения станка.

Указатель

\$

\$AA_ATOL, 579
\$AA_COUP_ACT
 при буксировке, 586
 при осевом соединении по главному
 значению, 611
\$AA_ESR_ENABLE, 708
\$AA_LEAD_SP, 610
\$AA_LEAD_SV, 610
\$AC_ACT_PROG_NET_TIME, 695
\$AC_ACTUAL_PARTS, 698
\$AC_AXCTSWA, 680
\$AC_AXCTSWE, 680
\$AC_CTOL, 579
\$AC_CUT_INV, 499
\$AC_CUTMOD, 498
\$AC_CUTMOD_ANG, 498
\$AC_CUTMODK, 498
\$AC_CUTTING_TIME, 695
\$AC_CYCLE_TIME, 695
\$AC_DELAYFST, 559
\$AC_ESR_TRIGGER, 708
\$AC_OLD_PROG_NET_TIME, 695
\$AC_OLD_PROG_NET_TIME_COUNT, 695
\$AC_OPERATING_TIME, 695
\$AC_OTOL, 579
\$AC_PROG_NET_TIME_TRIGGER, 696
\$AC_REPOS_PATH_MODE, 569
\$AC_REQUIRED_PARTS, 698
\$AC_SMAXVELO, 574
\$AC_SMAXVELO_INFO, 574
\$AC_SPECIAL_PARTS, 698
\$AC_TOTAL_PARTS, 698
\$AC_TRAFO_CORR_ELEM_P, 426
\$AC_TRAFO_CORR_ELEM_T, 426
\$AC_TRAFO_ORIAX_LOC, 426
\$AN_AXCTAS, 680
\$AN_AXCTSWA, 680
\$AN_ESR_TRIGGER, 708
\$AN_LANGUAGE_ON_HMI, 972
\$AN_POWERON_TIME, 694
\$AN_SETUP_TIME, 694
\$NT_CLOSE_CHAIN_T, 426
\$NT_CNTRL, 426
\$NT_CORR_ELEM_P, 425
\$NT_CORR_ELEM_T, 425
\$NT_NAME, 417

\$NT_ROT_AX_NAME, 494
\$NT_TRAFO_INDEX, 417
\$P_ACTBFRAME, 330
\$P_AD, 498
\$P_BFRAME, 330
\$P_CHBFRAME, 329
\$P_CHBFMASK, 330
\$P_CTOL, 579
\$P_CUT_INV, 499
\$P_CUTMOD, 498
\$P_CUTMOD_ANG, 498
\$P_CUTMOD_ERR, 500
\$P_CUTMODK, 498
\$P_DELAYFST, 559
\$P_IFRAME, 331
\$P_IS_EES_PATH, 234
\$P_NCBFRAME, 329
\$P_NCBFMASK, 330
\$P_ORI_DIFF, 489
\$P_ORI_POS, 489
\$P_ORI_SOL, 490
\$P_ORI_STAT, 493
\$P_OTOL, 579
\$P_PATH, 233
\$P_PFRAME, 331
\$P_PROG, 233
\$P_PROGPATH, 233
\$P_SIM, 297
\$P_STACK, 233
\$P_SUBPAR, 171
\$P_TOOLENV, 509
\$P_TOOLENVN, 508
\$PA_ATOL, 579
\$SA_LEAD_TYPE, 610
\$SC_CONTPREC, 552
\$SC_MINFEED, 552
\$SC_PA_ACTIV_IMMED, 247
\$SN_PA_ACTIV_IMMED, 247
\$TC_CARR1...14, 473
\$TC_CARR18...23, 474
\$TC_CARR18[m], 478
\$TC_DP1 ... 25, 427
\$TC_ECPxy, 432
\$TC_SCPxy, 432

*

* (функция вычисления), 81

/

/ (функция вычисления), 81

+

+ (функция вычисления), 81

<

< (оператор сравнения), 83

<< (связывающий оператор), 93

<= (оператор сравнения), 83

<> (оператор сравнения), 83

=

== (оператор сравнения), 83

>

> (оператор сравнения), 83

>= (оператор сравнения), 83

A

ABS, 81

ACCLIMA, 547

ACOS, 81

ACTBLOCNO, 183

ACTFRAME, 305

ADISPOSA, 299

ALF

для быстрого отвода от контура, 142

AND, 83

APR, 49

APRB, 49

APRP, 49

APW, 49

APWB, 49

APWP, 49

AS, 220

ASIN, 81

ASPLINE, 260

ATAN2, 81

ATOL, 576

AV, 619

AX, 671

AXCTSWE, 679

AXCTSWEC, 679

AXCTSWED, 679

AXIS, 31

AXNAME, 92

AXSTRING, 671

AXTOCHAN, 151

AXTOSPI, 671

A-сплайн, 267

B

B_AND, 83

B_NOT, 83

B_OR, 83

B_XOR, 83

BAUTO, 260

BFRAME, 305

BLOCK, 208

BLSYNC, 136

BNAT, 260

BOOL, 31

BOUND, 87

BRISK, 545

BRISKA, 545

BSPLINE, 260

BTAN, 260

B-сплайн, 268

C

CAC, 259

CACN, 259

CACP, 259

CALL, 207

CALLPATH, 210

CASE, 113

CDC, 259

CFINE, 315

Chan, 31

CHANDATA, 235

CHAR, 31

CHKDNO, 469

CIC, 259

CLEARARM, 129

CLRINT, 138

COARSE, 619

COARSEA, 299

COLLPAIR, 412

COMPCAD, 274

COMPCURV, 274

COMPLETE, 235
COMPOF, 274
COMPON, 274
COMPSURF, 274
CONTDCON, 730
CONTPRON, 724
CORRTRAFO, 419
COS, 81
COUPDEF, 619
COUPDEL, 619
COUPOF, 619
COUPOFS, 619
COUPON, 619
COUPONC, 619
COUPRES, 619
CP, 391
CPBC, 631
CPDEF, 630
CPDEL, 630
CPFMOF, 633
CPFMON, 633
CPFMSON, 632
CPFPOS + CPOF, 633
CPFPOS + CPON, 631
CPFRS, 631
CPLA, 630
CPLCTID, 631
CPLDEF, 630
CPLDEL, 630
CPLDEN, 631
CPLINSC, 635
CPLINTR, 635
CPLNUM, 631
CPLOF, 631
CPLON, 631
CPLOUTSC, 635
CPLOUTTR, 635
CPLPOS, 631
CPLSETVAL, 631
CPMALARM, 636
CPMBRAKE, 636
CPMPRT, 635
CPMRESET, 634
CPMSTART, 635
CPMVDI, 636
CPOF, 631
CPON, 630
CPRECOF, 552
CPRECON, 552
CPROT, 244
CPROTDEF, 239
CPSETTYPE, 636
CPSYNCOF, 636
CPSYNCOF2, 636
CPSYNCOV, 636
CPSYNFIP, 636
CPSYNFIP2, 636
CPSYNFIV, 636
CSPLINE, 260
CTAB, 599
CTABDEF, 588
CTABDEL, 595
CTABEND, 588
CTABEXISTS, 595
CTABFNO, 604
CTABFPOL, 604
CTABFSEG, 604
CTABID, 597
CTABINV, 599
CTABISLOCK, 597
CTABLOCK, 596
CTABMEMTYP, 597
CTABMPOL, 604
CTABMSEG, 604
CTABNO, 604
CTABNOMEM, 604
CTABPERIOD, 597
CTABPOL, 604
CTABPOLID, 604
CTABSEG, 604
CTABSEGID, 604
CTABSEV, 599
CTABSSV, 599
CTABTEP, 599
CTABTEV, 599
CTABTMAX, 599
CTABTMIN, 599
CTABTSP, 599
CTABTSV, 599
CTABUNLOCK, 596
CTOL, 576
CTRANS, 315
CUT3DC, 447
CUT3DCC, 457
CUT3DCCD, 457
CUT3DCD, 447
CUT3DF, 451
CUT3DFD, 451
CUT3DFF, 451
CUT3DFS, 451
CUTMOD, 495
CUTMODK, 495
CYCLE4071
внешнее программирование, 849

CYCLE4072
 внешнее программирование, 850
 CYCLE4073
 внешнее программирование, 854
 CYCLE4074
 внешнее программирование, 855
 CYCLE4075
 внешнее программирование, 858
 CYCLE4077
 внешнее программирование, 861
 CYCLE4078
 внешнее программирование, 864
 CYCLE4079
 внешнее программирование, 866
 CYCLE435 - установка системы координат
 правящего инструмента
 внешнее программирование, 808
 CYCLE495 - Профилирование
 внешнее программирование, 808
 CYCLE60 - гравирование
 внешнее программирование, 759
 CYCLE61 - плоское фрезерование
 внешнее программирование, 762
 CYCLE62 - вызов контура
 внешнее программирование, 764
 CYCLE63 - фрезерование кармана
 внешнее программирование, 765
 CYCLE64 - предварительное сверление кармана
 внешнее программирование, 767
 CYCLE70 - резьбофрезерование
 внешнее программирование, 769
 CYCLE72 - фрезерование траектории
 внешнее программирование, 770
 CYCLE76 - прямоугольная цапфа
 внешнее программирование, 774
 CYCLE77 - круговая цапфа
 внешнее программирование, 776
 CYCLE78 - Сверлильное резьбофрезерование
 внешнее программирование, 778
 CYCLE79 - многогранник
 внешнее программирование, 782
 CYCLE800 - поворот
 внешнее программирование, 810
 CYCLE801 - образец позиции "Решетка/рамка"
 внешнее программирование, 814
 CYCLE802 - любые позиции
 внешнее программирование, 815
 CYCLE81 - центрование
 внешнее программирование, 784
 CYCLE82 - сверление
 внешнее программирование, 785

CYCLE83 - глубокое сверление
 внешнее программирование, 787
 CYCLE830 - Глубокое сверление 2
 внешнее программирование, 818
 CYCLE832 - High Speed Settings
 внешнее программирование, 825
 CYCLE84 - нарезание внутренней резьбы без
 компенсирующего патрона
 внешнее программирование, 790
 CYCLE840 - нарезание внутренней резьбы с
 компенсирующим патроном
 внешнее программирование, 829
 CYCLE85 - развертывание
 внешнее программирование, 793
 CYCLE86 - растачивание
 внешнее программирование, 794
 CYCLE899 - Фрезерование открытого паза
 внешнее программирование, 831
 CYCLE92 - отрез
 внешнее программирование, 796
 CYCLE930 - выточка
 внешнее программирование, 834
 CYCLE940 - канавка
 внешнее программирование, 838
 CYCLE95 - съём припуска вдоль контура
 внешнее программирование, 797
 CYCLE951- обработка резаньем
 внешнее программирование, 840
 CYCLE952 — прорезание контура
 внешнее программирование, 843
 CYCLE98 - цепочка резьб
 внешнее программирование, 799
 CYCLE99 - нарезание резьбы резцом
 внешнее программирование, 803
 C-сплайн, 269

D

DCI, 54
 DCM, 54
 DCU, 54
 DEF, 31
 DEFAULT, 113
 DEFINE ... AS, 220
 DELAYFSTOF, 557
 DELAYFSTON, 557
 DELDL, 433
 DELETE, 158
 DELOBJ, 405
 DELTOOLENV, 506
 Disable, 138
 DISPLOF, 183

DISPLON, 183
 DISPR, 563
 DIV, 81
 dl, 431
 DO, 641
 DRIVE, 545
 DRIVEA, 545
 DV, 619
 DYNFINISH, 549
 DYNNORM, 549
 DYNPOS, 549
 DYNROUGH, 549
 DYNSEMIFIN, 549

E

Easy XML, 689
 EAUTO, 260
 EES, 225
 EG
 Электронный редуктор, 612
 EGDEF, 612
 EGDEL, 617
 EGOFC, 616
 EGOFs, 616
 EGON, 613
 EGONSYN, 613
 EGONSYNE, 613
 ELSE, 123
 Enable, 138
 ENAT, 260
 ENDFOR, 125
 ENDIF, 123
 ENDLABEL, 115
 ENDLOOP, 124
 ENDWHILE, 126
 ESR, 707
 ESRR, 713
 ESRS, 713
 ETAN, 260
 EVERY, 641
 EXECSTRING, 78
 EXECTAB, 736
 EXECUTE, 739
 exp, 81
 EXTCALL
 для SINUMERIK 828D, 215
 для SINUMERIK 840D sl, 212
 EXTCLOSE, 700
 EXTERN, 201
 EXTOPEN, 700

F

false, 31
 FCTDEF, 443
 FCUB, 540
 FENDNORM, 298
 FFWOF, 551
 FFWON, 551
 FIFOCTRL, 554
 FILEDATE, 162
 FILEINFO, 162
 FILESIZE, 162
 FILESTAT, 162
 FILETIME, 162
 FINE, 619
 FINEA, 299
 FLIN, 540
 FNORM, 540
 FOR, 125
 FPO, 540
 FRAME, 31
 FROM, 641
 FTOCOFC, 446
 FTOCON, 446

G

G290, 720
 G291, 720
 G5, 387
 G62, 298
 G621, 298
 G7, 387
 G810 ... G819, 297
 G820 ... G829, 297
 GEOAX, 674
 GET, 146
 GETACTTD, 471
 GETD, 146
 GETDNO, 470
 GETTCOR, 509
 GETTENV, 507
 GETVARAP, 66
 GETVARDFT, 69
 GETVARDIM, 68
 GETVARLIM, 67
 GETVARPHU, 66
 GETVARTYP, 70
 GOTO, 110
 GOTOB, 110

GOTOC, 110
GOTOF, 110
GOTOS, 109
GP, 75
GROUP_ADDEND — конец дополнения ввода
 внешнее программирование, 869
GROUP_BEGIN — Начало блока программы
 внешнее программирование, 868
GROUP_END — Конец блока программы
 внешнее программирование, 869
GUD, 32
G-группа
 Технология, 549
G-код
 запрограммировать косвенно, 74

Н

High Speed Settings — CYCLE832
 внешнее программирование, 825
HOLES1 - образец позиции "Линия"
 внешнее программирование, 744
HOLES2 - образец позиции "Окружность"
 внешнее программирование, 745

I

ID, 641
IDS, 641
IF, 123
IFRAME, 305
INDEX, 96
INICF, 31
INIPO, 31
INIRE, 31
INIT, 129
INITIAL, 235
INITIAL_INI, 235
INT, 31
INTERSEC, 734
IPOBRKA, 299
IPOENDA, 299
IPOSTOP, 619
IPTRLOCK, 560
IPTRUNLOCK, 560
ISAXIS, 671
ISFILE, 161
ISNUMBER, 92
ISOCALL, 209
ISVAR, 64

J

JERKLIM, 572
JERKLIMA, 547

L

L..., 199
LEAD, 349
LEADOF, 606
LEADON, 606
LENTOAX, 529
LIFTFAST, 140
Link
 -переменные, 29
LLI, 45
LN, 81
LONGHOLE - продольный паз
 внешнее программирование, 757
LOOP, 124
LUD, 32

M

M17, 187
M30, 187
MASLDEF, 638
MASLDEL, 638
MASLOF, 638
MASLOFS, 638
MASLON, 638
MATCH, 96
MAXVAL, 87
MCALL, 205
MD10010, 129
MD10280, 129
MD15800, 29
MD18104, 505
MD18116, 506
MD18156, 29
MD20360, 514
MD24558, 516
MD24658, 516
MEAC, 286
MEAFRAME, 323
MEAS, 283
MEASA, 286
MEAW, 283
MEAWA, 286
MINDEX, 96

MINVAL, 87
 MMC, 689
 MOD, 81
 MODAXVAL, 671
 MPF, 224

N

NAMETOINT, 408
 NCK, 31
 NEWCONF, 153
 NOC, 619
 NOT, 83
 NPROT, 244
 NPROTDEF, 239
 NUMBER, 92
 NUT, 360

O

OEMIPO1/2, 297
 OMA1 ... OMA5, 297
 OR, 83
 ORIAxes, 358
 ORIC, 463
 ORICONCCW, 360
 ORICONCW, 360
 ORICONIO, 360
 ORICONTO, 360
 ORICURVE, 364
 ORID, 463
 ORIEULER, 358
 ORIMKS, 356
 ORIPATH, 373
 ORIPATHS, 373
 ORIPLANE, 360
 ORIRESET(A, B, C), 347
 ORIROTA, 368
 ORIROTC
 при вращение ориентации инструмента, 368
 при интерполяции вращения инструмента, 374
 ORIROTR, 368
 ORIROTT, 368
 ORIRPY, 358
 ORIRPY2, 358
 ORIS, 463
 ORISOF, 381
 ORISOLH, 485
 ORISON, 381
 ORIVECT, 358
 ORIVIRT1, 358

ORIVIRT2, 358
 ORIWKS, 356
 OS, 643
 OSB, 643
 OSC, 463
 OSCILL, 649
 OSCTRL, 643
 OSD, 463
 OSE, 643
 OSNSC, 643
 OSOF, 463
 OSP1, 643
 OSP2, 643
 OSS, 463
 OSSE, 463
 OST, 463
 OST1, 643
 OST2, 643
 OTOL, 576

P

p..., 203
 P_ACTFRAME, 331
 PCALL, 210
 PDELAYOF, 657
 PDELAYON, 657
 PFRAME, 305
 PHI
 Полиномы ориентации, 367
 при ориентации вдоль боковой поверхности конуса, 360
 PHU, 46
 PL
 при полиномиальной интерполяции, 275
 при сплайн-интерполяции, 260
 po, 275
 PO[PHI]
 Полиномы ориентации, 366
 при вращение ориентации инструмента, 373
 при ориентации вдоль боковой поверхности конуса, 360
 PO[PSI]
 Полиномы ориентации, 366
 при вращение ориентации инструмента, 373
 при ориентации вдоль боковой поверхности конуса, 360
 PO[THT]
 Полиномы ориентации, 367
 при вращение ориентации инструмента, 373

PO[XH]

Полиномы ориентации, 367
при задаче ориентации двух контактных точек, 364

PO[YH]

Полиномы ориентации, 367
при задаче ориентации двух контактных точек, 364

PO[ZH]

Полиномы ориентации, 367
при задаче ориентации двух контактных точек, 364

POCKET3 - Прямоугольный карман
внешнее программирование, 747

POCKET4 - Круговой карман
внешнее программирование, 749

POLF

для управляемого ЧПУ отвода, 708

POLFA, 708

POLFMASK

для управляемого ЧПУ отвода, 708

POLFMLIN

для управляемого ЧПУ отвода, 708

POLY, 275

POLYPATH, 275

PON, 666

PONS, 657

POSFS, 619

POT, 81

PREPRO, 186

PRESETON, 319

PRESETONS, 321

Prio, 136

PRLOC, 31

Process DataShare, 700

PROTA, 413

PROTD, 415

PROTS, 414

PSI

Полиномы ориентации, 367
при ориентации вдоль боковой поверхности конуса, 360

PTP, 391

PTPG0, 391

PTPWOC, 391

PUD, 32

PUNCHACC, 657

PUTFTOC, 445

PUTFTOCF, 444

PW, 260

R

READ, 159

REAL, 31

REDEF, 38

RELEASE, 146

REP, 57

REPEAT, 115

REPEATB, 115

REPOSA, 563

REPOSH, 563

REPOSHA, 563

REPOSL, 563

REPOSQ, 563

REPOSQA, 563

RET, 188

RET (параметрируемая), 189

RETB (параметрируемая), 195

RG, 28

RINDEX, 96

RMBBL, 563

RMEBL, 563

RMIBL, 563

RMNBL, 563

ROUND, 81

ROUNDUP, 164

Run MyScreens, 689

R-параметры (параметры для расчета)

Глобальный, 28

Спец. для канала, 26

S

Save, 176

SBLOF, 178

SBLON, 178

SCPARA, 683

SD, 260

SD41610, 426

SD41611, 426

SD42475, 379

SD42476, 379

SD42477, 379

SD42900, 437

SD42910, 437

SD42920, 437

SD42930, 438

SD42935, 440

SD42940, 441

SD42984, 497

set, 57
SETAL, 705
SETDNO, 470
SETINT, 136
SETM, 129
SETTCOR, 516
SIN, 81
SLOT1 - Продольный паз
 внешнее программирование, 752
SLOT2 - кольцевая канавка
 внешнее программирование, 755
SOFT, 545
SOFTA, 545
SON, 657
SONS, 657
SPF, 224
SPI, 671
SPIF1, 657
SPIF2, 657
SPLINEPATH, 272
SPN, 662
SPOF, 657
SPP, 662
SPRINT, 99
SQRT, 81
START, 129
STARTFIFO, 554
STAT, 392
STOPFIFO, 554
stopre, 554
STRING, 31
STRINGIS, 685
STRLEN, 95
SUBSTR, 97
SYNR, 31
SYNRW, 31
SYNW, 31

T

TAN, 81
TANG, 533
TANGDEL, 538
TANGOF, 537
TANGON, 536
TCARR, 479
TCOABS, 479
TCOFR, 479
TCOFRX, 479
TCOFRY, 479
TCOFRZ, 479

THETA

 при вращение ориентации инструмента, 368
 при интерполяции вращения инструмента, 374

TILT, 349
TLIFT, 535
TMOF, 669
TMON, 669
TOFFOF, 482
TOFFON, 482
TOLOWER, 94
TOOLENV, 503
TOUPPER, 94
TOWBCS, 438
TOWKCS, 438
TOWMCS, 438
TOWSTD, 438
TOWTCS, 438
TOWWCS, 438
TRAANG

 с программируемым углом, 386

TRACON, 389
TRACYL, 383
TRAFOOF, 403
TRAFOON, 417
TRAILOF, 583
TRAILON, 583
Transmit, 383
TRAORI, 346
TRUE, 31
TRUNC, 81
Tu, 396

U

ULI, 45
UNTIL, 127

V

VELOLIM, 573
VELOLIMA, 547

W

WAITC, 619
WAITE, 129
WAITENC, 681
WAITM, 129
WAITMC, 129
WHEN, 641
WHEN-DO, 652

WHENEVER, 641
WHENEVER-DO, 652
WHILE, 126
WORKPIECE, 715
WRITE, 154

X

XML пользователя, 689
XOR, 83

A

Аварийные сообщения
 устанавливать в программе ЧПУ, 705
Аварийные сообщения циклов, 705
Автоматический указатель прерываний, 562
Автоматического разделения пути, 662
Адреса OEM, 297
Адресация, 227
Асинхронное качание, 643
Атрибуты позиций
 запрограммировать косвенно, 75

Б

Бесконечный цикл, 124
Блок программы
 -повторение, 115
Буксировка, 583
Буксируемые оси, 585
Быстрый отвод от контура, 140

В

Ведомая ось
 при осевом соединении по главному
 значению, 606
Ведущая ось
 при осевом соединении по главному
 значению, 606
Вектор направления, 352
Вектор ориентации THETA, 368
 внешнее программирование, 868
Внешнее смещение нулевой точки, 317
Вращение
 вектора ориентации, 368
Время обработки, 695
Вспомогательная точка фрезы, 450

Вывод
 на внешнее устройство/файл, 700
Вызов контура - CYCLE62
 внешнее программирование, 764
Вызов подпрограммы с указанием пути и
 параметрами, 210
Вырубка, 657
Выточка - CYCLE930
 внешнее программирование, 834

Г

Гео-ось
 переключить, 674
Глобальная память программ обработки деталей
(GDIR), 226
Глубина вложенности
 управляющих структур, 121
Глубина врезания, 450
Глубокое сверление - CYCLE83
 внешнее программирование, 787
Глубокое сверление 2 - CYCLE830
 внешнее программирование, 818
Гравирование - CYCLE60
 внешнее программирование, 759
Граничные условия для трансформаций, 402

Д

Данные окружности
 вычислить, 737
Движение "от точки к точке", 391
Движение "от точки к точке" в декартовой системе
 координат, 391
Деталь
 -главная директория, 224
 -директории, 224
 -счетчик, 698
Доступность
 зависящие от системы, 5

З

Замедление на внутренних углах, 298
Замедление на всех углах, 298
Защищенные области, 239
Значение износа, 432

И

Имя диска, 229
 Имя подпрограммы в коде DIN, 231
 Имя файла, 231
 Индекс поля, 59
 Индикация кадра
 подавить, 183
 Инициализация
 полей, 57
 Инструмент
 -ориентация, 463
 -ориентация при смене фрейма, 481
 -параметры, 427
 Инструментальный суппорт
 -кинематика, 473
 -ориентируемый, 479
 Интерполяция вектора вращения, 368

К

Кадр останова, 561
 Канавка - CYCLE940
 внешнее программирование, 838
 Качание
 Асинхронное, 643
 Асинхронное качание, 643
 Синхронное качание, 649
 управлять через синхронное действие, 649
 Частичная подача, 651
 Кинематика
 подвижная, 478
 Класс данных, 54
 Кольцевая канавка - SLOT2
 внешнее программирование, 755
 Компонент фрейма
 FI, 311
 tr, 311
 Конец блока программы — GROUP_END
 внешнее программирование, 869
 Конец дополнения ввода — GROUP_ADDEND
 внешнее программирование, 869
 Контур
 -кодирование, 730
 повторный подвод, 563
 -подготовка, 724
 -таблица, 724
 Коррекции на инструмент
 аддитивные, 431

Коррекции радиуса инструмента
 Замедление на углах, 298
 Коррекция длин инструмента Online, 482
 Коррекция инструмента
 Память коррекций, 427
 Система координат для значений износа, 438
 Косвенное программирование
 G-кодов, 74
 адресов, 72
 атрибутов позиций, 75
 строк программы обработки детали, 78
 Косоугольная трансформация (TRAANG)
 с программируемым углом, 386
 Коэффициент полинома, 276
 коэффициент связи, 583
 Критерий окончания движения
 программируемый, 299
 Круговая цапфа - CYCLE77
 внешнее программирование, 776
 Круговой карман - POCKET4
 внешнее программирование, 749

Л

Логические операторы, 83
 Любые позиции - CYCLE802
 внешнее программирование, 815

М

Макрос, 220
 Маршрут поиска
 при вызове подпрограммы, 232
 Программируемый маршрут поиска, 210
 Массив, 57
 Маятниковое движение
 Область возврата, 651
 Подача в точке возврата, 653
 Точка возврата, 651
 Метка, 115
 Метка перехода
 при переходах в программе, 111
 при повторении программных блоков, 115
 Многогранник - CYCLE79
 внешнее программирование, 782

Н

Нарезание внутренней резьбы без
 компенсирующего патрона - CYCLE84
 внешнее программирование, 790

Нарезание внутренней резьбы с
компенсирующим патроном - CYCLE840
 внешнее программирование, 829
Нарезание резьбы резцом - CYCLE99
 внешнее программирование, 803
Начало блока программы — GROUP_BEGIN, 868
не зависящий от регистра, 227
Номер D
 свободно присвоить, 469
Номер резца, 469
Номера D
 переименовать, 470
 проверить, 469

О

Обработка резаньем
 поддерживающие функции, 723
Обработка резаньем - CYCLE951
 внешнее программирование, 840
Обработчик прерываний
 Быстрый отвод от контура, 140
 Выключение/включение, 138
 Движение отвода, 142
 Новое согласование, 138
 Программируемое направление
 перемещения, 142
 Удаление, 138
Образец позиции "Линия" - HOLES1
 внешнее программирование, 744
Образец позиции "Окружность" - HOLES2
 внешнее программирование, 745
Образец позиции "Решетка/рамка" - CYCLE801
 внешнее программирование, 814
Округление, 164
Оперативная память, 235
Операторы сравнения, 83
Определение заготовки, 715
Ориентация инструмента
 относительно траектории, 371
Ориентация инструмента относительно
траектории, 371
Ориентируемые инструментальные
суппорта, 473
Осевое соединение по главному значению, 606
Оси
 Буксируемая, 585
Оси вращения
 Векторы направления, 473
 Векторы расстояния, 473
 Угол поворота, 473
Оси ориентации, 358

Оставшееся время
 для детали, 697
Останов
 автономный для привода, 713
 управляемый ЧПУ, 712
Острие фрезы, 450
Ось
 -переход, 146
Отвод
 автономный для привода, 713
 управляемый ЧПУ, 708
Отдельный кадр
 -подавление, 178
Отрез - CYCLE92
 внешнее программирование, 796

П

Память
 Оперативная-,
 Предварительная обработка-, 554
 Программная-, 223
Память коррекций, 427
Параметр
 -передача при вызове подпрограммы, 169,
 201
 Фактический-, 169
 Формальный-, 169
параметры
 Инструмент, 427
Первичная установка ориентации инструмента
ORIRESET, 348
Переключаемые гео-оси, 674
Переменная
 Преобразование типов, 90
Переменные
 определенные пользователем, 31
 Преобразование типа, 91
Переход
 на метки перехода, 110
 на начало программы, 109
Плоское фрезерование - CYCLE61
 внешнее программирование, 762
Поворот - CYCLE800
 внешнее программирование, 810
Повторение программного блока
 с помощью косвенного программирования
 CALL, 208
Подготовка контура
 Квитирование ошибок, 739
Подпрограмма
 -возврат параметризуемый (RET ...), 189

- возврат параметризуемый (RETB...), 195
- вызов без передачи параметров, 199
- вызов косвенный, 207
- вызов модальный, 205
- вызов с передачей параметров, 201
- имя, 166
- повторение, 203
- Применение, 165
- Программируемый маршрут поиска, 210
- Поле
 - определение, 57
 - элемент, 57
- Полином-знаменатель, 279
- Полиномиальная интерполяция, 275
- Полярная трансформация, 339
- Предварительная обработка
 - память, 554
- Предварительное сверление контурного кармана
 - CYCLE64
 - внешнее программирование, 767
- Представление EES, 227
- Представление NCK, 227
- Программа
 - Адресация, 227
 - ветвление, 113
 - время выполнения, 694
 - Инициализация, 235
 - память, 224
 - переходы, 110
 - повторение, 203
- Программа инициализации, 235
- Программирование ориентации, 358
- Программная память
 - Стандартные директории, 224
 - Типы файлов, 224
- Программный цикл
 - Конечный цикл, 124
 - Счетный цикл, 125
 - Цикл IF, 123
 - Цикл REPEAT, 127
 - Цикл WHILE, 126
- Продольный паз - LONGHOLE
 - внешнее программирование, 757
- Продольный паз - SLOT1
 - внешнее программирование, 752
- Прорезание контура — CYCLE952
 - внешнее программирование, 843
- Профилирование - CYCLE495
 - внешнее программирование, 808
- Прямоугольная цапфа - CYCLE76
 - внешнее программирование, 774

- Прямоугольный карман - POCKET3
 - внешнее программирование, 747
- Пусковое событие
 - при измерении, 290
- Путь директории, 230

Р

- Рабочий цикл
 - характеристика управляющих структур, 122
- Развертывание - CYCLE85
 - внешнее программирование, 793
- Растачивание - CYCLE86
 - внешнее программирование, 794
- Регистрация и поиск непригодных для поиска областей, 561
- Режим работы
 - при измерении, 291
- Режим ускорения, 545
- Резьбофрезерование - CYCLE70
 - внешнее программирование, 769
- Рывок
 - коррекция, 572
 - ограничение, 545

С

- Сверление - CYCLE82
 - внешнее программирование, 785
- Сверлильное резьбофрезерование - CYCLE78
 - внешнее программирование, 778
- Сглаживание
 - характеристики ориентации, 381
- Символ 0, 91
- Симуляция главного значения, 610
- Сингулярные позиции, 357
- Синхронное качание
 - Движение подачи, 653
 - Обработка такт IPO, 654
 - Определение подач, 652
 - Подача в области возврата, 653
 - Синхронные действия, 652
 - Следующая частичная подача, 655
 - Согласование качающейся оси и оси подачи, 652
- Синхронность по положению, 619
- Синхронность по положению с угловым смещением, 619
- Синхронность по частоте вращения, 619

Синхронный ход
 грубый, 622
 точный, 622
 Синхронный шпиндель
 -определить пару, 624
 Соединение, 619
 Система
 -зависящие, доступность, 5
 Системные переменные
 Ограничение измерительного щупа, 294
 Состояние измерительного щупа, 294
 Системные фреймы, 329
 Смещение нулевой точки
 Внешнее смещение, 317
 Соединение
 базовое, 630
 строк, 93
 Соединение по главному значению
 Синхронизация ведущей и ведомой оси, 608
 Соединение по фактическому и заданному значению, 609
 Соединение по заданному значению, 622
 Соединение по скорости, 622
 Соединение по фактическому значению, 622
 Соединение сплайнов, 272
 Состояние задания измерения, 294
 Состояние связи
 при буксировке, 586
 Состояние соединения
 при осевом соединении по главному значению, 611
 Сплайн
 -интерполяция, 260
 -типы,
 Стирание остатка пути, 292
 Строка
 -длина, 95
 -операции, 91
 -соединение, 93
 форматировать, 99
 Структура буксировок, 583
 Счетный цикл, 125
 Съём припуска вдоль контура - CYCLE95
 внешнее программирование, 797

Т

Тип кинематики, 478
 Тип соединения, 622
 Типы трансформаций
 Общая функция, 335
 Торцовое фрезерование, 355

Точка обработки фрезы, 450
 Точность контура
 программируемая, 552
 Трансформации
 Кинематические трансформации, 336
 Независящая от кинематики первичная установка ориентации инструмента, 336
 Последовательная связь трансформаций, 337
 связанные, 389
 Трансформация ориентации, 335
 Трех-, четырех- и пятиосевая трансформация, 346
 Трансформация боковой поверхности цилиндра, 339
 Трансформация ориентации TRAORI
 Базовая 5/6-осевая трансформация, 339
 Варианты программирования ориентации, 347
 Движения ориентации, 337
 Кинематика станка, 338
 Программирование ориентации, 347
 Трансформация с качающейся линейной осью, 345

У

Угол RPY, 352
 Угол Эйлера, 352
 Указание пути, 229
 Управляющие
 -структуры, 121
 Установка системы координат правящего инструмента - CYCLE435
 внешнее программирование, 808
 Установочное значение, 432

Ф

Файл
 -информация, 162
 Фрезерование кармана – CYCLE63
 внешнее программирование, 765
 Фрезерование открытой канавки - CYCLE899
 внешнее программирование, 831
 Фрезерование траектории - CYCLE72
 внешнее программирование, 770
 Фрейм
 вызывать, 312
 -цепочка, 333
 Фрейм-компонент
 MI, 311

RT, 311

sc, 311

Фрейм-переменная

Вызов трансформаций координат, 303

Предопределенные фрейм-переменные, 305

Присваивание значений, 309

Фреймы

Глобальные ЧПУ, 327

присвоить, 313

Системный-, 329

Спец. для канала, 328

Цепочки фреймов, 313

Функции OEM, 297

Ц

Центрование - CYCLE81

внешнее программирование, 784

Цепочка резьб - CYCLE98

внешнее программирование, 799

Цепь детали, 424

Цепь инструмента, 424

Ш

Шлифование с врезанием с угловой подачей, 387

Шпиндель

-переход, 146

Штамповка, 657

Э

Электронный редуктор, 612

Элемент контура

обработать, 736

Я

Языковой режим, 720

